

PCT/JP2005/001559

日 本 国 特 許 庁
JAPAN PATENT OFFICE

27.1.2005	
REC'D 17 FEB 2005	
WIPO	PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 4 年 1 月 2 7 日
Date of Application:

出 願 番 号 特 願 2 0 0 4 - 0 1 8 5 2 4
Application Number:
[ST. 10/C]: [J P 2 0 0 4 - 0 1 8 5 2 4]

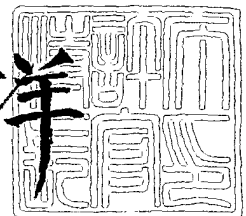
出 願 人 日 本 電 気 株 式 会 社
Applicant(s):

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2 0 0 4 年 1 0 月 8 日

特許庁長官
Commissioner,
Japan Patent Office

小 川 洋



出証番号 出証特 2 0 0 4 - 3 0 9 1 0 5 8

【書類名】 特許願
【整理番号】 35001230
【提出日】 平成16年 1月27日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/445
【発明者】
 【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内
 【氏名】 小比賀 亮仁
【特許出願人】
 【識別番号】 000004237
 【氏名又は名称】 日本電気株式会社
【代理人】
 【識別番号】 100088959
 【弁理士】
 【氏名又は名称】 境 廣巳
【手数料の表示】
 【予納台帳番号】 009715
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9002136

【書類名】 特許請求の範囲**【請求項 1】**

OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報をOS内に復元することを特徴とする高速再起動方法。

【請求項 2】

OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させると共に、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定し、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元することを特徴とする高速再起動方法。

【請求項 3】

OSの再起動前に、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報をOS内に復元することを特徴とする高速再起動方法。

【請求項 4】

OSの再起動前に、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存してあるプロセスIDテーブルを参照して、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報をOS内に復元することを特徴とする高速再起動方法。

【請求項 5】

ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新し、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にし、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元することを特徴とする高速再起動方法。

【請求項 6】

或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定することを特徴とする請求項2または5記載の高速再起動方法。

【請求項 7】

主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージからOSの起動を行うことを特徴とする請求項1乃至5の何れか1項に記載の高速再起動方法。

【請求項 8】

システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域への

アクセスに変換することを特徴とする請求項7記載の高速再起動方法。

【請求項 9】

OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする情報処理装置。

【請求項 10】

OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定する再起動フラグ設定手段と、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする情報処理装置。

【請求項 11】

OSの再起動前に、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする情報処理装置。

【請求項 12】

前記プロセス情報退避手段は、OSの再起動前に、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存するプロセスIDテーブルを参照して、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるものであることを特徴とする請求項11記載の情報処理装置。

【請求項 13】

ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス退避領域作成手段と、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新するプロセス退避情報更新手段と、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にするプロセス情報退避領域解放手段と、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする情報処理装置。

【請求項 14】

或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定する手段を備えることを特徴とする請求項10または13記載の情報処理装置。

【請求項 15】

主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージからOSの起動を行う手段を備えることを特徴とする請求項9乃至13の何れか1項に記載の情報処理装置。

【請求項 16】

システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換する手段を備えることを特徴とする請求項15記載の情報処理装置。

【請求項 17】

コンピュータを、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とするプログラム。

【請求項 18】

コンピュータを、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定する再起動フラグ設定手段、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とするプログラム。

【請求項 19】

コンピュータを、OSの再起動前に、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とするプログラム。

【請求項 20】

前記プロセス情報退避手段は、OSの再起動前に、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存するプロセスIDテーブルを参照して、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるものであることを特徴とする請求項19記載のプログラム。

【請求項 21】

コンピュータを、ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス退避領域作成手段、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新するプロセス退避情報更新手段、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にするプロセス情報退避領域解放手段、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とするプログラム。

【請求項 22】

コンピュータを、さらに、或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定する手段、として機能させることを特徴とする請求項18または21記載のプログラム。

【請求項 23】

コンピュータを、さらに、主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージからOSの起動を行う手段、として機能させることを特徴とする請求項17乃至21の何れか1項に記載のプログラム。

【請求項 2 4】

コンピュータを、さらに、システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換する手段、として機能させることを特徴とする請求項23記載のプログラム。

【書類名】明細書

【発明の名称】高速再起動方法および情報処理装置ならびにプログラム

【技術分野】

【0001】

本発明は、APサーバーや汎用情報処理装置などに好適な高速再起動方法に関し、特に、再起動前に運用していたアプリケーションをそのまま再起動後も継続して運用できる高速再起動方法および情報処理装置ならびにプログラムに関する。

【背景技術】

【0002】

再起動は、OSの設定を変更した場合、設定を反映させるために利用される。Windows(登録商標)OSの場合は、レジストリの内容を変更し、変更を反映させるために再起動が必要になる。さらに、システムに何らかの障害が発生し、発生した障害の原因を特定することが困難である場合は、いち早く障害からシステムを回復させる最初的手段として、再起動を用いることが多い。このような再起動は、システムを運用している間は頻繁に利用するものであり、システムの停止時間を最小限にとどめるため、再起動処理は高速化が望まれていた。

【0003】

再起動処理の高速化を実現するための従来の技術の例としては、特許文献1に記載される情報処理装置の高速再起動方法がある。図25はこの従来の情報処理装置の構成を示したものである。図25に示すように従来の情報処理装置1は、その中に主記憶装置11が配置され、外部には入出力装置2がある。また、主記憶装置11の内部にはオペレーティングシステム(以下、OSと呼ぶ)がロードされるOSプログラム領域111、初期化データ領域112、非初期化データ領域113、初期化データ退避領域114が配置されており、主記憶装置11以外の情報処理装置内部は、再起動開始手段12、高速再起動指定手段13、実行制御手段14、初期化データ退避手段15、初期化データ回復手段16、入出力停止手段17、OSロード手段18から構成されている。

【0004】

この従来の情報処理装置1では、OS起動時にOSの初期化データを初期化データ退避手段15により初期化データ退避領域114に退避し、高速再起動指定手段13により高速再起動が指定されると、実行制御手段14によりOSプログラムの実行開始番地に実行を移し、初期化データ回復手段16により初期化データを回復する。以上の処理を行うことで、OSのプログラムおよびデータを再ロードすることなく情報処理装置を高速に再起動する高速再起動方式を提供している。

【特許文献1】特開平11-24936号公報(第7頁、図1)

【発明の開示】

【発明が解決しようとする課題】

【0005】

しかしながら、特許文献1に記載された従来の高速再起動方法では、OSの高速再起動を実現することはできるが、OS上で実行されているアプリケーション(アプリケーションプログラム)については言及されていない。OSを再起動する際にはアプリケーションを正常終了または強制終了させる必要がある。再起動時間を短縮するためにアプリケーションを強制終了させると、再起動後の当該アプリケーションの正常運用を保障できない。なぜなら、アプリケーションの強制終了は、現在のサービス状態を保存しない。アプリケーションによっては、運用時のサービス状態が再起動後に必要な場合がある。そのような場合には再起動後にアプリケーションの正常運用を保障することができなくなる。よって、アプリケーションを正常終了させる必要があるが、アプリケーションの正常終了は、現在のサービス状態を保存するために、ディスクなどの処理速度の遅い補助記憶装置へのアクセスが頻繁に発生する。このような補助記憶装置へのアクセスは、システムの終了処理時間を増大させることとなり、システムの再起動時間を増大させるものとなる。

【0006】

また、再起動を利用する際に考慮すべき事項として、次のような場合が想定される。再起動は主にシステムの設定をOSに反映させたり、あるアプリケーションの障害によりOSが正常に動作しなくなった場合に利用される。複数のアプリケーションを使ってサービスを提供するサーバにおいて、ある一つのアプリケーションに障害が発生し、OSが正常に動作しなくなった場合は、再起動する必要があるが、その際には、障害の発生したアプリケーション以外のアプリケーションも終了させる必要があり、サービス途中にあるアプリケーションはサービスを強制中断せざるを得なくなる。

【0007】

上記の問題点を克服するために、本発明では、シャットダウン時の処理を最小限にとどめて、システムを高速に再起動できる高速再起動方法および情報処理装置ならびにプログラムを提供することを目的とする。

【0008】

本発明の他の目的は、再起動後もアプリケーションを継続運用できる高速再起動方法および情報処理装置ならびにプログラムを提供することを目的とする。

【課題を解決するための手段】

【0009】

課題を解決するための手段を説明するにあたり、本発明を理解するための要素技術の説明を行う。

【0010】

計算機はプログラムを実行する。計算機中のプログラムはプロセスという実行単位に分割されて実行される。ある時点において計算機が実行できるプロセスは一つであるので、プロセスを複数同時に実行するには、当該プロセスを管理する上位プログラムが必要となる。それがOSにあたるが、OSは複数のプロセスを管理するために、OS自身が利用している主記憶領域上にプロセス表というものを持っている。プロセス表はプロセス一つにつき一つ用意され、プロセスを実行するための情報や、実行途中の情報を格納している。よって、あるプロセスは当該プロセスの利用している主記憶領域とプロセス表を保存しておくことにより、実行途中にあるプロセスを中断したり、中断した時点から処理を再開することができる。一つまたは複数のプロセスによってアプリケーションが構成される。そこで本発明では、アプリケーションの実行時などにOS内に存在するプロセス表やその他、プロセスを復元するために必要な情報を主記憶領域上の退避領域に複写しておき、再起動時には、アプリケーションの使用していた主記憶の領域はそのままに、OSのみを初期化する。そして初期化の後に、退避しておいたプロセス情報(プロセス表やその他OSが管理していた情報)をOSのプロセス管理領域に上書きするよう動作する。このような構成を採用することで、再起動前のプロセスを再起動後も継続実行できる環境を提供することができる。より具体的には、本発明は以下のような構成を有する。

【0011】

本発明の第1の高速再起動方法は、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報をOS内に復元することを特徴とする。

【0012】

本発明の第2の高速再起動方法は、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させると共に、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定し、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元することを特徴とする。

【0013】

本発明の第3の高速再起動方法は、OSの再起動前に、OS再起動後に継続運用するユーザ

プロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報をOS内に復元することを特徴とする。ここで、OS再起動後に継続運用するユーザプロセスかどうか、あるいはその反対にOS再起動時に再起動するユーザプロセスかどうかは、例えば、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存してあるプロセスIDテーブルを参照して、判断することができる。

【0014】

本発明の第4の高速再起動方法は、ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させ、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新し、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にし、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元することを特徴とする。

【0015】

本発明の第5の高速再起動方法は、第2または第4の高速再起動方法において、或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定することを特徴とする。

【0016】

本発明の第6の高速再起動方法は、第1乃至第4の何れかの高速再起動方法において、OSの使用していた主記憶領域の初期化は、主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージをOSの使用していた主記憶領域に上書きすることで行うことを特徴とする。

【0017】

本発明の第7の高速再起動方法は、第6の高速再起動方法において、システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換することを特徴とする。

【0018】

本発明の第1の情報処理装置は、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする。

【0019】

本発明の第2の情報処理装置は、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定する再起動フラグ設定手段と、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする。

【0020】

本発明の第3の情報処理装置は、OSの再起動前に、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段と、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする。ここで、プロセス情報退避手段は、例えば、OSの再起動前に、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存するプロセスIDテーブルを参照して、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるものである。

【0021】

本発明の第4の情報処理装置は、ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス退避領域作成手段と、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新するプロセス退避情報更新手段と、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にするプロセス情報退避領域解放手段と、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段と、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段とを備えることを特徴とする。

【0022】

本発明の第5の情報処理装置は、第2または第4の情報処理装置において、或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定する手段を備えることを特徴とする。

【0023】

本発明の第6の情報処理装置は、第1乃至第4の何れかの情報処理装置において、前記主記憶初期化手段は、OSの使用していた主記憶領域の初期化は、主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージをOSの使用していた主記憶領域に上書きすることで行うものであることを特徴とする。

【0024】

本発明の第7の情報処理装置は、第6の情報処理装置において、システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換する手段を備えることを特徴とする。

【0025】

本発明の第1のプログラムは、コンピュータを、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とする。

【0026】

本発明の第2のプログラムは、コンピュータを、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定する再

起動フラグ設定手段、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とする。

【0027】

本発明の第3のプログラムは、コンピュータを、OSの再起動前に、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス情報退避手段、OSの再起動時に、前記ユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とする。ここで、プロセス情報退避手段は、例えば、OSの再起動前に、OSの再起動後に継続運用するプロセスまたは継続運用しないプロセスの識別子を保存するプロセスIDテーブルを参照して、OS再起動後に継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるものである。

【0028】

本発明の第4のプログラムは、コンピュータを、ユーザプロセスの生成時、前記生成したユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させるプロセス退避領域作成手段、前記ユーザプロセスの切替え時、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグを設定するとともに、再起動しない場合には退避領域に退避されているプロセス情報を最新の状態に更新するプロセス退避情報更新手段、前記ユーザプロセスの終了時、前記退避させたプロセス情報を無効にするプロセス情報退避領域解放手段、OSの再起動時に、前記再起動フラグが再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化する主記憶初期化手段、OS再起動後に、前記退避したプロセス情報のうち前記再起動フラグが再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元するプロセス復元手段、として機能させることを特徴とする。

【0029】

本発明の第5のプログラムは、第2または第4のプログラムにおいて、コンピュータを、さらに、或るユーザプロセスに関するプロセス情報に再起動するかどうかを指定する再起動フラグを設定する際、そのユーザプロセスと同じユーザアプリケーションプログラムに属する他の全てのユーザプロセスを検索し、該検索した全てのユーザプロセスに関するプロセス情報中の再起動フラグも同じ値に設定する手段、として機能させることを特徴とする。

【0030】

本発明の第6のプログラムは、第1乃至第4の何れかのプログラムにおいて、前記主記憶初期化手段は、OSの使用していた主記憶領域の初期化は、主記憶装置の一部を構成する不揮発性記憶部に記憶されたOSの主記憶イメージをOSの使用していた主記憶領域に上書きすることで行うものであることを特徴とする。

【0031】

本発明の第7のプログラムは、第6のプログラムにおいて、コンピュータを、さらに、システム運用中にOSから前記不揮発性記憶部への書き込みアクセスが発生する毎に、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを前記不揮発性記憶部から主記憶装置の一部を構成する読み出し書き込み可能な主記憶部に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換する手段、として機能させることを特徴とする。

【発明の効果】

【0032】

第1の効果は、システムの終了時間を削減できることにある。その理由は次の通りである。システムの終了処理とは、OSの終了処理とアプリケーションの終了処理から構成され

る。OSは通常終了することになるが、アプリケーションの終了処理に関しては、当該アプリケーションを構成するプロセスのプロセス表などのプロセス情報を主記憶上の退避領域に退避させているだけである。よって、アプリケーションの終了処理に伴う転送速度の遅い補助記憶装置に対するアクセスを軽減することになるので、システムの終了時間を短縮することができる。

【0033】

第2の効果は、再起動後も再起動前に運用していたアプリケーションを継続実行できることにある。その理由は、再起動前にOSの使用している主記憶上よりプロセス表などのプロセス情報を退避し、再起動後にはOSの使用している主記憶上にプロセス表などのプロセス情報を復元するため、プロセス表を退避、復元したプロセスは、実行を中断、再開したに過ぎず、あたかも継続運用されているように見えるためである。よって、障害の発生していないアプリケーションは、サービスを継続提供することができる。

【発明を実施するための最良の形態】

【0034】

次に、本発明の第1の実施の形態について図面を参照して詳細に説明する。

【0035】

図1を参照すると、本発明の第1の実施の形態にかかる情報処理装置100は、プログラム制御により動作するCPU(プロセッサ)110と、主記憶領域120と、OS130から構成されている。プロセスはCPU110により実行される。主記憶領域120は、OSプログラム領域121、OSデータ領域122、プロセス使用領域123、退避領域124を含み、退避領域124内には実行されているプロセス分だけプロセス情報退避領域125が存在する。OS130は、プロセス情報退避手段131、プロセス復元手段132、主記憶初期化手段133、再起動フラグ設定手段134を含む。

【0036】

図2に退避領域124のデータ構造を示す。退避領域124には、プロセス情報退避領域125とよばれる個々のプロセスごとにOSの管理している情報を退避させる領域があり、プロセス情報退避領域125は単方向リスト205でつながれている。さらに、退避領域使用サイズ201という退避領域124の大きさを示すデータ領域を持つ。OSの管理している情報とは、プロセス表202とブロックテーブル203である。OSは1つのプロセスにつき1つのプロセス表を作成して保守している。プロセス表には、プロセスの状態、プログラムのカウンタやスタックポインタ、メモリの割り当て、オープンしているファイルの状態やスケジュール情報、プロセスに関するあらゆる情報が入っている。また、OSは主記憶領域120をブロックという単位で管理しており、ブロックテーブル203とは、個々のプロセスに一つ存在し、当該プロセスの使用しているブロックのアドレスが記述されている。再起動フラグ204は、当該プロセスを再起動させるかどうかを判断するためのフラグである。本実施の形態では、プロセス表202とは独立に再起動フラグ204を設けてあるが、プロセス表202の中に再起動フラグを持つようにしてもよい。

【0037】

退避領域124は、基本的に主記憶領域120の一番後方のアドレスから始まり、主記憶領域120の先頭に向かって、現在実行しているプロセスの数だけ主記憶領域120を使用する。退避領域124を後方の定位置からアドレッシングする方法は、次のような利点がある。OSは、起動の際に主記憶領域の管理を行うために、主記憶領域の初期化を行う。本発明において主記憶領域初期化の際にOSは、再起動前に使用されていた退避領域124と、プロセス自身がプログラムを実行するために使用していたプロセス使用領域123を知る必要がある。再起動前にプロセスによって使用されていたプロセス使用領域123を起動後に知るには、定位置から退避領域124を置くとよい。起動後にOSは、退避領域124がどこにあるのかを調べる必要がなく、OSプログラムの一部として「プロセス情報退避領域125は主記憶領域120の一番後方のアドレスから置かれる」と記述することで、プロセス情報退避領域125がどこにあるのかを知ることができ、退避領域124のプロセス情報退避領域125からプロセス使用領域123を知ることができる。よって、再起動前の退避領域124の位置がわかりさえすれば退避領域124の主記憶領域120上での位置は一番後方でなくてもよく、また別の記憶装置

(フラッシュメモリなどの不揮発性主記憶装置)に退避領域124を保存しておいてもよい。

【0038】

プロセス情報退避手段131、プロセス復元手段132、主記憶初期化手段133、再起動フラグ設定手段134は、それぞれ次のような機能を有する。

【0039】

プロセス情報退避手段131は、OS内部にあるプロセス表とブロックテーブルをプロセス情報退避領域125に退避させる。

【0040】

プロセス表の退避処理とは、あるプロセスについて、プロセス表をプロセス情報退避領域125へOSデータ領域122からコピーすることである。

【0041】

通常、OSは主記憶領域120を複数のブロックに分割して管理しており、OSはプロセスに必要な量のブロックを与えることで、プロセスを実行している。どのブロックを当該プロセスが使用しているかという情報はブロックテーブルに保存されている。ブロックテーブルの退避処理とは、プロセスに付随するブロックテーブル203を退避領域124にOSデータ領域122からコピーすることである。

【0042】

再起動フラグ設定手段134は、ユーザからの入力によって、あるプロセス情報退避領域125内に設けられた再起動フラグ204を設定する手段である。ユーザからの入力は、プロセスを特定するためのプロセスIDである。再起動フラグ204を設定されたプロセスは、再起動されるため、再起動後に再起動前の状態に復元せず、当該プロセスのプロセス情報退避領域125と、プロセス使用領域123は解放される。これに対して、再起動フラグ204を設定されないプロセスは、再起動されず、当該プロセスのプロセス情報退避領域125とプロセス使用領域123を用いて、再起動前の状態に復元される。

【0043】

主記憶初期化手段133は、OSが起動前に実行していたプロセスについて、当該プロセスが利用していた領域を認識するための処理である。通常、OSは起動時に利用できる主記憶領域120を認識するために、ブロックごとに初期化処理を行う。ここでプロセス情報退避領域125より、起動前に実行していたプロセスが使用しているブロックをブロックテーブル203より検索し、使用しているブロックについては初期化処理を施さないようにする。

【0044】

プロセス復元手段132は、プロセス情報退避領域125に格納されているプロセス表202、ブロックテーブル203からプロセスを復元する処理である。復元処理とは、OSデータ領域122にプロセス表202およびブロックテーブル203を書き戻す(コピー)処理である。

【0045】

次に、図1ないし図5を参照して本実施の形態について全体の動作を詳細に説明する。

【0046】

最初に、システム終了時までの処理について説明する。まず、全てのプロセスについて、プロセス情報退避手段131を用いてプロセス情報退避領域125を作成する(図3S100)。この処理は全てのプロセスについて行い、全てのプロセスについてプロセス情報退避領域125が作成できたとき終了する(S101)。

【0047】

プロセス情報退避領域125の作成は、まず主記憶上にプロセス情報退避領域125を確保し、当該プロセスのプロセス表202をOSデータ領域122から読み出して、確保したプロセス情報退避領域125にコピーし、次にOSデータ領域122からブロックテーブル203を読み出して、確保したプロセス情報退避領域125にコピーすることで行う。また、作成時には退避領域使用サイズ201を当該プロセスのために使用した主記憶領域分だけ足す。さらに、必要に応じて再起動フラグ204を設定し、単方向リスト205でプロセス情報退避領域125どうしをつなげる。

【0048】

こうして全てのプロセスについてプロセス情報退避領域125の作成を行った後、OSの通常終了処理を行う(S102)。プロセスについては終了処理をせず、そのままの状態(OS起動後のプロセスの復元処理を待つ。プロセス情報退避領域125の作成は、システム終了直前、つまりユーザがシステム終了命令を発行してから作成を開始し、OSの終了処理を始めるまでの間に作成するが、任意の時間に作成してもよい。

【0049】

再起動フラグ設定手段134による再起動フラグ204の設定は、システム終了までの任意の時間に行う。再起動フラグ204は、プロセス情報退避領域125の中に設置されており、当該プロセスの退避領域が作成されるまで設定することができないので、当該プロセスの退避領域が作成される前にユーザから再起動する旨の指示があれば、OSは一時的に当該プロセスに対する再起動フラグ設定の情報を保持しておく必要がある。その際の設定の情報は、再起動フラグ設定手段134の中にプロセスIDのリストとして保持しておく。その後プロセス情報退避領域125が作成されたら、以前のユーザの指示に従い、再起動フラグ204を設定する。

【0050】

次に、起動時の処理を説明する。OSの展開を含むOSの起動(S103)については、LinuxなどのOSと同様である。Linuxの展開については、例えば参考文献1「D. Bovet and M. Cesati . 詳解Linuxカーネル, オライリー・ジャパン, 2001.」に記載されている。起動時の処理の説明は、OSの主記憶上への展開が終了し、OSが最初の処理を行うところから始める。

【0051】

まず、主記憶領域120の利用可能領域を認識するためにOSにより主記憶の初期化がブロック単位で行われる(S104)。この初期化の際には再起動前にプロセスによって使用されていた領域をOSに認識させる必要がある。OSはあるブロックが使用中であるか否かを認識するために、ブロック識別子400を持つ(図4)。ブロック識別子400は、ブロック番号401と使用フラグ402とから構成される。ブロック識別子400は一つのブロックについて一つ存在する。起動時には、OSデータ領域122にブロック識別子400のための領域を確保し、使用フラグ402を零(使用されていない)に設定することにより主記憶領域120の初期化とみなされる。利用可能である主記憶領域の開始アドレスは、あらかじめBIOS などによってその値が与えられている。

【0052】

図5は本発明における主記憶初期化手段133のフローチャートを示している。本発明における主記憶初期化手段133を実行する前にあらかじめ退避領域124の使用している領域を確認する(S201)。退避領域124の使用している領域は、退避領域124にある退避領域使用サイズ201を見ることで確認することができる。次に、退避領域124内にあるすべてのプロセス情報退避領域125を検索し、再起動フラグ204の設定されていない(再起動前のプロセスを継続利用する)プロセスのブロックテーブル203からあらかじめ初期化の必要のないブロックを把握しておく(S202)。退避領域124によって使用されているブロックのブロック番号401のリストを一時的に保持しておき、のちの処理に使用する。

【0053】

退避領域使用サイズ201を確認し、初期化の必要のないブロックを把握した上で、指定されたアドレスに対するブロック識別子一つ分の主記憶領域を0で初期化する(S204)。次に、退避領域124で先ほど初期化したブロックが使用されているかを確認する(S205)。使用しているなら(S206のYES)、使用フラグを立て(S207)、次のアドレスへ移る。このようにして、利用可能である領域の全てのブロック識別子400を初期化すると(S203のYES)主記憶初期化手段133を終了する。

【0054】

主記憶領域120の初期化が終了すると、次にプロセスを復元する。退避領域124内のプロセス情報退避領域125を確認し、再起動フラグ204が設定されているか確認する(S107)。再起動を行うのであれば(S108)、当該プロセス情報の退避領域124を解放し、次のプロセスの退避情報に関して処理を行う(S105)。再起動を行わないのであれば、プロセスを復元す

る(S109)。以上の処理を退避領域124内にあるプロセス情報退避領域125のエントリが全てなくなるまで行う(S106)。プロセスの復元は、プロセス表とブロックテーブルのための領域をOSデータ領域122上に確保し(S110、S112)、退避しておいたプロセス表202とブロックテーブル203を用意した領域にコピーする(S111、S113)。あるプロセス情報の退避領域124の解放時には、以下の処理を行う。

【0055】

1) 削除するプロセス情報退避領域の一つ前のプロセス情報退避領域(単方向リスト205の一つ前のリスト要素)の単方向リスト205が指す次のプロセス情報退避領域へのポイントを、削除するプロセス情報退避領域に格納されている単方向リスト205の内容に置き換える。

2) 削除するプロセス情報退避領域が使用している主記憶ブロックのブロック識別子400内の使用フラグ402を零(使用していない)にする。

3) 退避領域使用サイズ201の値から、削除するプロセス情報退避領域分だけ差し引く。

【0056】

次に、本実施の形態の効果について説明する。

【0057】

本実施の形態では、プロセス(アプリケーション)の終了処理を省略し、プロセスの終了にかかる補助記憶装置などの転送速度の遅い装置への入出力処理をなくすることができるので、システムの再起動時間を短縮することができる。

【0058】

さらに、任意のプロセスを再起動後も再起動前の状態を保持したまま継続実行することができる。よって、システムの再起動に際してプロセスが提供していたサービスを強制的に中断する必要がなくなる。

【0059】

次に、本発明の第1の実施の形態の変形例について図面を参照して詳細に説明する。

【0060】

図6を参照すると、本発明の第1の実施の形態の変形例にかかる情報処理装置100aは、再起動フラグ設定手段134の代わりに再起動プロセスIDテーブル134aと再起動設定手段134bを備えている点などで、図1に示した本発明の第1の実施の形態の情報処理装置100と相違する。

【0061】

本発明の第1の実施の形態においては、OSの再起動前に、ユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域に退避させると共に、前記退避させたプロセス情報に再起動するかどうかを指定する再起動フラグ204を設定し、OSの再起動時に、再起動フラグ204が再起動しない設定になっているユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し、OS再起動後に、前記退避したプロセス情報のうち再起動フラグ204が再起動しない設定になっているユーザプロセスのプロセス情報をOS内に復元した。

【0062】

これに対して、第1の実施の形態の変形例では、再起動するユーザプロセスのIDを保持する再起動プロセスIDテーブル134aを設け、再起動設定手段134bが、少なくともOSの終了前に、ユーザから入力されるプロセスIDに基づいて、再起動するユーザプロセスのIDを再起動プロセスIDテーブル134aに設定する。プロセス情報退避手段131は、図7のフローチャートに示すように、OSの再起動前に、OSデータ領域122から一つのプロセス表を取り出し(S110)、そのプロセス表に設定されたプロセスIDが再起動プロセスIDテーブル134aに登録されているかどうかを調べることにより(S111)、当該プロセス表に対応するユーザプロセスが再起動するプロセスか、OS再起動後に継続運用するユーザプロセスかを判断し(S112)、再起動せずに継続運用するユーザプロセスに関するOS内のプロセス情報を主記憶装置上の退避領域124に退避させる(S100)。すなわち、図2に示したようなプロセス情報退避領域125を作成する。但し、本変形例の場合、再起動フラグ204は省略してよい。そして、OSの

再起動時(S102、S103)、主記憶初期化手段133により、前記退避したユーザプロセスの使用していた主記憶領域は初期化せずにOSの使用していた主記憶領域は初期化し(S104)、前記退避したプロセス情報をOS内に復元する(S105、S106、S109)。

【0063】

なお、上記の説明では、再起動プロセスIDテーブル134aに、再起動するユーザプロセスのプロセスIDを設定したが、その反対に、再起動せずに継続運用するユーザプロセスのプロセスIDを設定するようにしてもよい。

【0064】

第1の実施の形態では、再起動の必要なプロセスについてもプロセス情報退避領域125を作成していたが、本変形例では、再起動の必要のないプロセスのみプロセス情報退避領域125を作成するので、第1の実施の形態よりも、プロセス情報退避領域125を作成する回数を減らすことができる。

【0065】

次に、本発明の第2の実施の形態について図面を参照して詳細に説明する。

【0066】

図8を参照すると、本発明の第2の実施の形態にかかる情報処理装置100bは、再起動フラグ設定手段134の代わりに、再起動プロセスIDテーブル134aと再起動設定手段134bを備え、プロセス情報退避手段131の代わりに、プロセス退避情報作成手段135、プロセス退避情報更新手段136及びプロセス情報退避領域解放手段137を備えている点などで、図1に示した本発明の第1の実施の形態と相違する。ここで、再起動プロセスIDテーブル134aと再起動設定手段134bは、本発明の第1の実施の形態の変形例におけるものと同様のものである。つまり、再起動設定手段134bは、ユーザから入力されるプロセスIDに基づいて、再起動するユーザプロセスのIDを再起動プロセスIDテーブル134aに設定する。

【0067】

本実施の形態では、本発明の第1の実施の形態におけるプロセス情報退避手段131を3つの手段に分割する。第1の実施の形態では、任意の時間にプロセス情報退避手段131を用いてプロセスの退避情報を作成していたが、本実施の形態では、プロセス生成時、プロセス切り替え時、プロセス終了時にそれぞれ、プロセス情報退避領域を作成し、更新し、領域を解放する。これら三つの処理をそれぞれプロセス退避情報作成手段135、プロセス退避情報更新手段136、プロセス情報退避領域解放手段137で実行する。プロセス切り替えとは、以下のような処理を指す。UNIX（登録商標）のようなマルチタスクOSでは、プロセスは複数同時に実行されるが、ある微小時間 Δt でCPUは一つのプロセスしか実行することができない。よってプロセスがあたかも複数同時に実行しているように見せかけるために、マルチタスクOSでは、プロセスの実行を一定時間ごとに切り替えている。これをプロセス切り替えという。以下では、プロセス情報退避領域の作成、更新、領域解放について説明する。

【0068】

図7はプロセス生成時に動作するプロセス退避情報作成手段135のフローチャートを示している。プロセス退避情報作成手段135は、OSによるプロセスの生成が完了した時点で処理を開始する。プロセスの生成完了とは、生成するプロセスのプロセス表202をOSデータ領域122に用意し、当該プロセスにプロセス使用領域123を割り当て、その使用領域123に実行するプログラムをコピーし、当該プロセスが実行準備を完了した時点を目指す。まず、プロセス情報退避領域125を主記憶領域120上に確保する(図9のS301)。次に、当該プロセスのプロセス表202をOSデータ領域122からプロセス情報退避領域125にコピーする(S302)。次に、当該プロセスのブロックテーブル203をOSデータ領域122からプロセス情報退避領域125にコピーする(S303)。最後に、単方向リスト205を設定し、退避領域124のリストに登録する(S304)。

【0069】

図10はプロセス切り替え時に動作するプロセス退避情報更新手段136のフローチャートを示している。まず該当するプロセスのプロセス情報退避領域125を退避領域124から検索

する(S401)。次に、ユーザから当該プロセスを再起動する旨の指示があるかどうかを再起動プロセスIDテーブル134aを参照してチェックする(S402)。ユーザからの指示により再起動するのであれば(S403)、前記検索したプロセス情報退避領域125における再起動フラグ204を立て(S404)、プロセス退避情報更新手段136を終了する。再起動せず継続運用するのであれば(S403)、再起動フラグ204を取り消し(S405)、OSデータ領域122上の最新のプロセス表202をプロセス情報退避領域125にコピー(S406)し、OSデータ領域122上の最新のブロックテーブル203をプロセス情報退避領域125にコピー(S407)し、プロセス退避情報更新手段136を終了する。ユーザからの指示が存在しない場合は、再起動フラグに従う、再起動フラグ204が立っている状態であるのなら、更新処理(S406、S407)は行わず終了する。

【0070】

図11はプロセス終了時に動作するプロセス情報退避領域解放手段137のフローチャートを示している。まず退避領域124から当該プロセスのプロセス情報退避領域125を検索する(S501)。次に単方向リスト205から当該プロセスのプロセス情報退避領域125を削除する(S502)。この処理(S502)は、当該プロセスのプロセス情報退避領域の前につながっている領域と後につながっている領域のリストをつなぎ合わせる処理になる。次に、当該プロセスのプロセス情報退避領域125を解放する(S503)。つまり、削除するプロセス情報退避領域125が使用している主記憶ブロックのブロック識別子400内の使用フラグ402を零(使用していない)にし、退避領域使用サイズ201の値から、削除するプロセス情報退避領域分だけ差し引く。これでプロセス情報退避領域解放手段137の処理が終了し、この後、OSによるプロセスの終了処理に移ることになる。

【0071】

再起動後の主記憶初期化、プロセス復元などは、第1の実施の形態と同様である。

【0072】

本実施の形態における効果は以下の通りである。

【0073】

第1の実施の形態では、プロセスの最新の情報を保持するためにはシステム終了時にプロセス情報退避領域125を作成する必要があった。例えば、システム終了10分前にあるプロセスについてプロセス情報退避領域125を作成すると、プロセス情報退避領域125を作成してからシステム終了までの10分間の間にプロセスは処理を実行し、プロセス表を書き換えたり、必要のなくなったブロックを解放し、ブロックテーブルを書き換えている可能性がある。このような場合にはプロセス復元手段132にてプロセス表202、ブロックテーブル203を元に戻しても実行できない場合がある。よって、プロセスの最新の情報を保持するためにシステム終了時にプロセス情報退避領域125を作成する必要があった。しかし、本実施の形態を適用すると、常にプロセス情報を最新の状態にしておくことができ、システム終了時にプロセスの終了処理を一切行う必要がなくなる。よって、第1の実施の形態と比較し、より高速にシステムを終了することができる。

【0074】

次に、本発明の第3の実施の形態について図面を参照して詳細に説明するが、その前に、第3の実施の形態に利用する主記憶イメージを用いた高速起動方式(以下主記憶イメージ方式と呼ぶ)について説明する。

【0075】

図12を参照すると、主記憶イメージ方式を適用した情報処理装置は、少なくとも主記憶装置101とデータ処理装置150とを有し、プログラム制御により動作する。

【0076】

主記憶装置101は、主記憶部126と不揮発性記憶部140とを含んで構成される。

【0077】

不揮発性記憶部140は、OS及びアプリケーションプログラムの初期化処理後の主記憶イメージを記憶する部分である。この不揮発性記憶部140は、不揮発なメモリなら任意の種類のメモリで良く、例えばフラッシュRAM等の書き換え可能な不揮発性メモリや、ROM等の読み出し専用の不揮発性メモリが使用できる。使用する主記憶イメージの内容を変更する

方法としては、不揮発性記憶部140の差し替え、または、不揮発性記憶部140が書き換え可能な場合は、ネットワークを利用したイメージファイルのダウンロードや、外部記憶媒体からのイメージファイルの読み込みなどの方法が使用可能である。

【0078】

不揮発性記憶部140に記憶する主記憶イメージの生成方法は任意であるが、例えば、通常のシステム起動方法で起動した直後の主記憶イメージをダンプすることにより生成することができる。

【0079】

主記憶部126は、不揮発性記憶部140以外の主記憶領域を構成する部分である。この主記憶部126は、読み書き可能なメモリなら任意の種類のメモリで良く、一般的にはDRAM、SRAM等の揮発性のメモリが使用される。但し、MRAM(Magnetoresistive RAM)、FeRAM(Ferroelectric RAM)、フラッシュRAM等の不揮発性のメモリを使用することも可能である。

【0080】

他方、データ処理装置150は、マッピングテーブル変更部190とアドレス変換部200とハードウェア初期化部180の機能手段を備え、ハードウェア初期化部210はマッピングテーブル初期化部180を有している。またデータ処理装置150は、制御テーブルとしてマッピングテーブル部160および保護情報部170を有している。

【0081】

マッピングテーブル部160は、データ処理装置150上で動作するOSやアプリケーションプログラム(両者を総称してプログラムと言う)が主記憶装置101を操作する際に指定するメモリアドレス(論理アドレス)を、主記憶装置101を構成する主記憶部126および不揮発性記憶部140に実際に割り当てられているメモリアドレス(物理アドレス)に変換するための情報、および保護モードを記憶する制御テーブルである。ここで、論理アドレスは論理ブロックアドレスとブロック内アドレスとで構成され、物理アドレスは物理ブロックアドレスと前記ブロック内アドレスとで構成される。

【0082】

図13にマッピングテーブル部160のエントリの構成例を示す。マッピングテーブル部160は、このようなエントリの集合から構成される。図13を参照すると、1つのエントリには、論理ブロックアドレス161と物理ブロックアドレス162と保護モード163の組が保持される。論理ブロックアドレス161は、論理アドレス空間を所定の大きさのブロックに分割したときにできる個々の論理ブロックを一意に識別するアドレスである。物理ブロックアドレス162は、物理アドレス空間を論理アドレス空間と同じ大きさのブロックに分割したときにできる個々の物理ブロックを一意に識別するアドレスである。ブロックとしては、ページと呼ばれるアーキテクチャに固有の単位を用いることで効率良く処理することが可能となるが、主記憶イメージ方式はその単位に制限されるわけではない。保護モード163は、読み込み専用の保護モードと読み書き可能な保護モードの何れか一方の値をとる。この保護モード163は、当該情報処理装置で実行されるプログラムの本来の保護属性とは異なる擬似的なものであり、不揮発性記憶部140のブロックに関してはシステム起動時に読み込み専用の保護モードが一律に設定される。

【0083】

保護情報部170は、当該情報処理装置で実行されるプログラム本来の保護属性を保持する制御テーブルである。図14に保護情報部170のエントリの構成例を示す。保護属性部170は、このようなエントリの集合で構成される。図14を参照すると、1つのエントリには、論理ブロックアドレス171と保護属性172の組が保持される。保護属性172には、読み込み専用の保護属性と、読み書き可能な保護属性とがあり、何れの保護属性が設定されるかは、論理ブロックアドレス171で特定されるブロックの本来の保護属性によって決定される。

【0084】

ハードウェア初期化部210は、当該情報処理装置の起動時および再起動時にシステム各部のハードウェアの初期化を行う部分であり、マッピングテーブル初期化部180を備えて

いる。このマッピングテーブル初期化部180は、主記憶装置101を構成する主記憶部126および不揮発性記憶部140への物理アドレスの割り当て機能と、マッピングテーブル部160の初期設定機能とを有する。一般にPCIバスの制御下に置かれているメモリなどのデバイスに対しては、BIOSの機能で物理アドレスを物理的に固定させずに或る程度動的に割り当てることができるが、マッピングテーブル初期化部180は、このようなBIOSの処理と同様な物理アドレスの割り当て機能を有している。但し、主記憶イメージ方式において物理アドレスの動的な割り当ては必須の事項でなく、物理アドレスが静的に割り当てられている計算機システムに対しても適用可能である。マッピングテーブル部160の初期設定では、論理アドレス空間を構成する各論理ブロックに対して、不揮発性記憶部140に割り当てた物理ブロックを順次に対応付け、それら全てのブロックの保護モードを、そのブロックの本来の保護属性にかかわらず読み出し専用に設定する。

【0085】

アドレス変換部200は、当該情報処理装置上で動作するプログラムがメモリ操作の際に指定する論理アドレスを物理アドレスに変換する機能と保護モード違反による例外を検出し例外処理を起動する機能とを有する。論理アドレスから物理アドレスへの変換および保護モード違反による例外検出は、マッピングテーブル部160を参照して行う。マッピングテーブル変更部190は、アドレス変換部200で起動された例外処理を実行する機能を有する。例外処理では、保護情報部170の参照による本来の保護属性の判定、不揮発性記憶部140のブロックの主記憶部126へのコピー、マッピングテーブル部160の更新などの処理が行われる。

【0086】

図15にアドレス変換部200およびマッピングテーブル変更部190の処理の一例を示す。アドレス変換部200は、プログラムから指定されたメモリアクセスにかかる論理アドレス中のブロックアドレスでマッピングテーブル部160を検索し、図13に示した論理ブロックアドレス161が一致するエントリを取得する(ステップS701)。次に、アドレス変換部200は、プログラムから指定されたメモリアクセスが、書き込みアクセスか、読み出しアクセスかを判別し(ステップS702)、読み出しアクセスであれば(ステップS702でNO)、ステップS701で取得したエントリ中の物理ブロックアドレス162を用いて、プログラムから指定された論理アドレスを物理アドレスに変換し(ステップS703)、その物理アドレスで特定される主記憶装置101の個所をアクセスする(ステップS704)。

【0087】

プログラムから指定されたメモリアクセスが書き込みアクセスの場合(ステップS702でYES)、アドレス変換部200は、ステップS701で取得したエントリ中の保護モード163が読み込み専用の保護モードか、読み書き可能な保護モードかを判別する(ステップS705)。読み書き可能な保護モードであれば(ステップS705でNO)、アドレス変換部200は、ステップS701で取得した物理ブロックアドレス162を用いて、プログラムから指定された論理アドレスを物理アドレスに変換し(ステップS703)、その物理アドレスで特定される主記憶装置101の個所をアクセスする(ステップS704)。

【0088】

保護モード163が読み込み専用の保護モードである場合(ステップS705でYES)、アドレス変換部200は、当該メモリアクセスの処理を一時保留し、例外イベントをマッピングテーブル変更部190に通知して例外処理を起動する(ステップS706)。この例外イベントでは、ステップS701で取得したエントリ中の論理ブロックアドレス161と物理ブロックアドレス162をマッピングテーブル変更部190に通知する。

【0089】

マッピングテーブル変更部190は、通知された論理ブロックアドレス161で保護情報部170を検索し、図14の論理ブロックアドレス171が一致するエントリの保護属性172を取得する(ステップS707)。次に、マッピングテーブル変更部190は、保護属性172が読み込み専用の保護属性か、読み書き可能な保護属性かを判別する(ステップS708)。読み込み専用の保護属性であれば(ステップS708でYES)、マッピングテーブル変更部190はアドレス変換部17

0に対してエラーを通知し、アドレス変換部200は保留中のメモリアクセスを拒否する等のエラー処理を行う。

【0090】

他方、保護属性172が読み書き可能な保護属性であれば(ステップS708でN0)、マッピングテーブル変更部190は、アドレス変換部200から通知された物理ブロックアドレス162で特定される不揮発性記憶部140のブロックを、主記憶部126の空きブロックにコピーする(ステップS709)。次にマッピングテーブル変更部190は、図13の論理ブロックアドレス161がアドレス変換部200から通知された論理ブロックアドレスに一致するエントリをマッピングテーブル部160から検索し、そのエントリの物理ブロックアドレス162をステップS709におけるコピー先の主記憶部126のブロックの物理ブロックアドレスに書き換え、かつ保護モード163を読み書き可能な保護モードに書き換える(ステップS710)。そして、例外処理の完了をアドレス変換部200に通知する。アドレス変換部200は、マッピングテーブル変更部190から例外処理の完了通知を受けると、一時保留していたメモリアクセスの処理をステップS701から再度開始する。

【0091】

図16は主記憶イメージ方式を適用した情報処理装置の概略動作を示すフローチャートである。以下、各図を参照して主記憶イメージ方式を適用した情報処理装置の動作を説明する。

【0092】

情報処理装置は起動されると、図16に示されるように、先ず、ハードウェア初期化部210によってハードウェアの初期化を行った後(ステップS801)、マッピングテーブル初期化部180によって、主記憶部126および不揮発性記憶部140に物理アドレスを割り当て、マッピングテーブル部160を初期設定する(ステップS802)。マッピングテーブル部160の初期設定では、論理ブロックアドレスと物理ブロックアドレスの対応関係をマッピングテーブル部160に書き込み、且つ、不揮発性記憶部140の全てのブロックは読み込み専用の保護モードでマッピングテーブル部160に登録する。

【0093】

次にデータ処理装置150は、不揮発性記憶部140に保存されている主記憶イメージからシステムを起動する(ステップS803)。この主記憶イメージは、OSやアプリケーションの起動直後と同じ状態であるため、システムは立ち上げ直後の環境に復帰することになる。このとき、主記憶イメージから起動してから、システム運用(ステップS804)が可能になるまでの運用中断状態は存在しない。

【0094】

システム運用(ステップS804)の段階において、プログラムから不揮発性記憶部140のとあるブロックに対して初めての書き込みアクセスが発生したときには、そこは読み込み専用の保護モードでマッピングされているので、アドレス変換部200が例外イベントを通知し、当該書き込みアクセスを一時保留する(図15のS706)。例外イベントを受け取ったマッピングテーブル変更部190は、保護情報部170を参照してその書き込み発生論理アドレスのブロックの保護属性を判断し(図15のステップS708)、読み書き可能属性であれば、そのブロックの大きさの空きブロックを主記憶部126から割り当て、その不揮発性記憶部140の当該ブロックの内容をコピーする(図15のステップS709)。更に、マッピングテーブル変更部190は、割り当てられた主記憶ブロックを書き込みが発生した論理ブロックアドレスに読み書き可の保護モードでマッピングするようマッピングテーブル部160に設定して、一時保留された書き込みアクセスの処理を再開させる(図15のステップS710)。この結果、一時保留された書き込みアクセスおよびその後の当該ブロックへの書き込み操作は、割り当てられた主記憶ブロックに対して実行される。このとき、ブロックの単位を小さくすることでコピー時間が短くなり、システム運用が中断される時間を短くすることができる。

【0095】

この状態で、再起動が行われると(図16のステップS805)、再びハードウェアの初期化処理(ステップS801)から実行され、起動時と同じようにマッピングテーブル初期化部180の

動作によってマッピングテーブル部160の内容が初期化され、不揮発性記憶部140がマッピングし直される(ステップS802)。このため、システムの運用中に主記憶部126に対して行われた書き込みの内容は一切反映されていない状態に戻り、これがつまり、立ち上げ直後の環境に復帰することと同じになる。

【0096】

このように主記憶イメージ方式を適用した情報処理装置にあっては、不揮発性記憶部140に、OS及びアプリケーションプログラムの初期化処理後の主記憶イメージが記憶されており、データ処理装置150は、不揮発性記憶部140に記憶された主記憶イメージからシステムを起動するため、システムの高速起動が可能になる。また、システム運用中にOSまたはアプリケーションプログラムから不揮発性記憶部140への書き込みアクセスが発生すると、データ処理装置150は、書き込みアクセスが発生したアドレスを含む所定幅のアドレス範囲内のデータを不揮発性記憶部140から主記憶部126に確保した代替領域にコピーし且つ以後の前記所定幅のアドレス範囲へのアクセスを前記代替領域へのアクセスに変換する設定を行う。このため、前記書き込みアクセスおよびそれ以降の前記所定幅のアクセス範囲へのアクセスは、前記代替領域へのアクセスに変換され、代替領域上のコピーが更新される。このように、OSやアプリケーションプログラムによって書き換える必要のあるデータをシステムの起動時に不揮発性記憶部140から読み書き可能な主記憶部126へ連続してコピーしておくのではなく、書き込みアクセスが発生した都度実施することにより、システム起動後に直ちにシステムの運用を開始することができる。

【0097】

以上が主記憶イメージを用いた高速再起動方式の説明である。続いて、この高速起動方式を応用した本発明の第3の実施の形態について図面を参照して説明する。

【0098】

図17を参照すると、本発明の第3の実施の形態にかかる情報処理装置100cは、主記憶領域120が主記憶部126と不揮発性記憶領域140で構成され、データ処理装置150が追加されている点で、図1に示した第1の実施の形態の情報処理装置100と構成が相違する。

【0099】

不揮発性記憶部140には、図12の不揮発性記憶部140の場合と異なり、OSのみの起動イメージだけが記憶されている。つまり、OSを通常起動した直後の主記憶上の内容を記憶している。不揮発性記憶部140に記憶されるOSの主記憶イメージ(起動イメージ)には、OSプログラム領域とOSデータ領域とが含まれる。主記憶部126内にもOSデータ領域122が図示されているが、このOSデータ領域122は不揮発性記憶部140内のOSデータ領域122が運用中に書き込みのためにアクセスされたことによって、図15のステップS709でコピーされて生成された領域であり、前述した「代替領域」に相当する。

【0100】

データ処理装置150は、図12で説明したような構成を備え、プログラムを実行する際のアドレス変換などを担う。

【0101】

本実施の形態では、第1の実施の形態の動作を説明した図3のフローチャートのOS起動部分(S103)に、OSの主記憶イメージを用いた高速起動方式を用いる。また、本実施の形態では、主記憶初期化手段133の処理内容が第1の実施の形態と異なる。すなわち、第1の実施の形態と同様、OSに対し既に使用されているブロックを認識させる必要があるが、主記憶イメージは、すでに主記憶領域の初期化を終了したものとみなすことができる。よって、第1の実施の形態における主記憶領域初期化手段133とは異なり、ブロック識別子400のための領域を主記憶上に確保する(S202)が必要がなく、すでに存在するブロック識別子に関して使用されているか否かを確認するのみでよい。また、第1の実施の形態と同様に、退避領域使用サイズ201とプロセスが使用しているブロックのリストは事前に知っておく必要がある。

【0102】

図18に本実施の形態における主記憶初期化手段133のフローチャートを示す。第1の実施

の形態との違いは、ブロック識別子のための主記憶領域を用意し 0 で初期化するか(S204)、すでに存在するブロック識別子を検索するか(S904)である。

【0103】

本実施の形態によれば、OSの起動にOS起動イメージを利用するため、OS起動時の初期化処理および転送速度の遅い補助記憶装置からの入出力をなくすることができるので、第1の実施の形態よりもさらに高速な再起動を実現することができる。勿論、常に再起動されるのはOSだけであり、再起動前に実行中のアプリケーションは、再起動フラグの値に応じて、再起動されるか、継続運用されるかが決定される。

【0104】

本実施の形態は、第1の実施の形態に主記憶イメージによる高速起動方式を適用したものであるが、第1の実施の形態の変形例(図6)または第2の実施の形態に主記憶イメージによる高速起動方式を適用し、OSの再起動の高速化を図るようにすることもできる。

【0105】

次に、本発明の第4の実施の形態について図面を参照して詳細に説明する。

【0106】

図19を参照すると、本発明の第4の実施の形態にかかる情報処理装置100dは、図1の第1の実施の形態にかかる情報処理装置100と比較して、再起動フラグ設定手段134に代えて、再起動設定受付手段220、再起動プロセスIDテーブル221及び再起動フラグ設定手段222を備えている点で相違する。

【0107】

再起動設定受付手段220は、ユーザから再起動するプロセスのプロセスIDをキーボード等を通じて受け付け、再起動プロセスIDテーブル221に保存する手段である。再起動フラグ設定手段222は、第1の実施の形態における再起動フラグ設定手段134と同様に、プロセス情報退避手段131によって退避領域124に生成されたプロセス情報退避領域125に設けられた再起動フラグ204の設定を、システム終了までの任意の時間に行う手段であるが、再起動プロセスIDテーブル221にプロセスIDが保存されたユーザ指定の再起動プロセスだけでなく、その再起動プロセスと同じアプリケーションに属するすべてのプロセスの再起動フラグ204をサーチし、これらのプロセスの再起動フラグ204も同様に設定する機能を有する。

【0108】

一般に、複数のプロセスによってアプリケーションが構成される場合、それら複数のプロセスには、同じプロセスグループIDが付与され、プロセス表202の1エントリに保存される。従って、或るプロセスと同じアプリケーションに属する他のプロセスがどれであるかは、プロセス表202に保存されているプロセスグループIDを調べることで可能である。

【0109】

図20に再起動フラグ設定手段222の処理のフローチャートを示す。まず再起動フラグ設定手段222は、再起動プロセスIDテーブル221にプロセスIDが保存されている1つのプロセスに注目し(ステップS1001)、そのプロセスに対応するプロセス情報退避領域125の再起動フラグ204をアクセスし、立っていないならば立てる(ステップS1002)。次に、注目中のプロセスのプロセス表202からプロセスグループIDを取得し(ステップS1003)、他のプロセス情報退避領域125のプロセス表202のプロセスグループIDを調べて、同じプロセスグループIDを持つ他のプロセスすべてを検索する(ステップS1004)。そして、検索したすべてのプロセスのプロセス情報退避領域125の再起動フラグ204をアクセスし、立っていないならば立てる(ステップS1005)。次に、再起動プロセスIDテーブル221にプロセスIDが保存されている他の1つのプロセスに注目を移し(ステップS1006)、ステップS1002に戻って上述した処理と同様の処理を繰り返す。再起動プロセスIDテーブル221に保存されたすべてのプロセスIDに注目した処理を終えた時点で(ステップS1007でYES)、処理を終える。

【0110】

このように本実施の形態によれば、ユーザは、アプリケーションを構成する複数のプロセスのうち少なくとも1つのプロセスを再起動プロセスとして指定しておけば、残りのプ

プロセスを再起動プロセスとして明示的に指定しなくても、自動的に再起動プロセスとして処理される。通常のアプリケーションは複数のプロセスから構成されることが多く、1つのアプリケーションを構成する複数のプロセスをこのように一括管理して再起動の設定を施すことで、個々のプロセスごとに再起動の設定を施すという煩雑な作業からユーザを解放することができる。また、再起動後にアプリケーションを構成するプロセスが足りない(アプリケーションの中の1つのプロセスのみ再起動してしまった。)などというミスを防ぐことができる。

【0111】

本実施の形態は、第1の実施の形態を前提としたが、同様の考えを第2の実施の形態あるいは第3の実施の形態に対して適用することが可能である。

【実施例】

【0112】

次に本発明の第1の実施例を、図面を参照して説明する。かかる実施例は本発明における第1の実施の形態に対応するものである。

【0113】

本実施例は具体的なOSの例として、Linuxを使用する。

【0114】

Linuxにおいて、プロセスを管理するためにカーネルが持っている情報としては、プロセスディスクリプタと、このプロセスディスクリプタに付随する他の構造体がある。プロセスディスクリプタは、プロセスを実行するために必要なほとんどの情報を保持している。他の構造体とは、例えばファイルをオープンしていれば、ファイル構造体、ソケットをオープンし、他の計算機と通信を行えば、ソケット構造体がプロセスディスクリプタに付随する。プロセスディスクリプタには、これら付随する構造体へのポインタが格納されている。これらプロセスディスクリプタと付随する構造体が、図2のプロセス表202にあたる。

【0115】

また、Linuxでは、主記憶領域120をページという単位で管理している。プロセスの使用している主記憶領域120は複数のページから構成され、複数のページをまとめて管理するためにプロセス1つにつき1つのページテーブルがOSにより用意される。このページテーブルが、ブロックテーブル203にあたる。

【0116】

プロセスを途中で中断し、再起動後にプロセスを継続実行するには、上記のプロセスディスクリプタ、ページテーブル、プロセスの使用メモリ領域のほかに、当該プロセスが実行中であれば、CPU内のレジスタに格納されているデータが必要となる。マルチタスクOSではプロセスを複数同時に実行するためにプロセス切り替えを行っている。

【0117】

プロセス切り替えによって、実行を一時中断されたプロセスのCPU内のデータは、タスク状態セグメントとカーネルモードスタックに分割して格納される。タスク状態セグメントは、プロセスディスクリプタの中にあるエントリの一つであり、カーネルモードスタックは、プロセスが使用しているメモリ領域の一部である。プロセスを再開するときは、タスク状態セグメントと、カーネルモードスタックから中断されたときのCPUレジスタの状態を復元し実行を再開する。

【0118】

プロセス情報退避手段131では、上記に記載したプロセスに関する情報(プロセスディスクリプタ、ページテーブルなど)をプロセス情報退避領域125に退避させることである。図21はプロセス情報退避手段131のLinuxでの処理を表している。

【0119】

生成された全てのプロセスのためのプロセスディスクリプタは、リスト構造になっており、プロセスディスクリプタの先頭アドレスは、init_task という変数に収められている。まず、init_task からプロセスディスクリプタのアドレスを取得する(ステップS1301)

。次にプロセス情報退避領域125を主記憶上に確保し(ステップS1302)、プロセスディスクリプタを作成したプロセス情報退避領域125にコピーする(ステップS1303)。同様に、プロセスディスクリプタに付随する他の構造体、ページテーブルもプロセス情報退避領域125にコピーする(ステップS1304、S1305)。プロセス情報退避領域125を単方向リスト205に登録し(ステップS1306)、退避領域使用サイズ201に使用した主記憶サイズを足す(ステップS1307)。

【0120】

以上でプロセス情報退避手段131の一度の処理が終了する。この処理を生成されたプロセス全てについて実行する。プロセス全てについてプロセス情報退避領域125を作成したのちLinuxを通常終了させる。

【0121】

システム起動時は主記憶初期化とプロセス復元の二つの処理を行う。

【0122】

Linuxでは、主記憶をページという単位で管理している。ページにはページディスクリプタとよばれる当該ページに関する情報を格納している領域があり、起動時にLinuxでは、Linux自身とハードウェアによって使用されている領域を除き、すべての主記憶領域120をページ単位に分割し、その各々にページディスクリプタを割り当てることで、利用できる主記憶領域120の初期化を行う。このような処理は、主記憶領域120をある単位(ページやセグメントなど)で管理しているOSに共通する処理である。主記憶初期化処理の時点で、再起動を行わないプロセスについては、当該プロセスが使用している主記憶空間をOSに認識させる必要がある。本実施例におけるページは、第1の実施の形態におけるブロックに相当し、ページディスクリプタはブロック識別子400に相当する。

【0123】

Linuxは、上述したとおり主記憶領域120をページという単位で管理しており、主記憶上のページの一つ一つにページディスクリプタが存在する。Linuxでは、このページディスクリプタの初期化作業をもって、主記憶領域120の初期化としている。Linuxの実装では、ページディスクリプタは、OS予約領域と呼ばれるOSをメモリ上に展開するために用意された領域の直後に配置され、固定長である。よって、再起動後もページディスクリプタを配置している主記憶上の位置、大きさは変わらない。また、主記憶領域120のページが使用されているかどうかは、ページディスクリプタ内のcount変数によって管理されている。count変数が1以上の時に当該ページは使用されていることになる。共有メモリを使用する複数のプロセスは、1つのページを複数のプロセスで使用するようになる。このような場合には、count変数は1以上の値をとることになる。

【0124】

図22に本実施例における主記憶初期化手段133の処理を示す。まず、退避領域124の使用している主記憶領域120を退避領域使用サイズ201から確認しておく(ステップS1401)。次に退避領域124内にある全てのプロセス情報退避領域125内のページテーブルを検索し、どのページがどのプロセスによって利用されているかどうかを調べておく(ステップS1402)。調べた結果は、ページディスクリプタ番号と使用しているプロセス数のテーブルにして保持しておく。

【0125】

主記憶領域120の最初のアドレス(厳密には、OSの予約領域の直後のアドレス)から、ページディスクリプタを設置していく。指定されたアドレスにつき一つのディスクリプタ領域を用意し0で初期化する(ステップS1404)。次に、指定されたアドレスが退避領域124に格納されているページテーブルで使用されているかを確認する(ステップS1405)。確認には上記にて調べたページディスクリプタ番号と使用しているプロセス数のテーブルを利用する。使用していれば(ステップS1406のYES)、ページディスクリプタ内のcount変数に使用しているプロセス数を代入する(ステップS1407)。使用していない場合(ステップS1406のNO)もしくはステップS1407の処理を終了すると、次のページディスクリプタの初期化に移る。利用できるすべての主記憶領域120を初期化すると(ステップS1403のYES)、主記憶

初期化手段133の処理が完了する。

【0126】

プロセス復元手段132は、プロセスディスクリプタおよび付随する他の構造体ならびにページテーブルを復元のために用意したOSデータ領域122にコピーすることであり、実施の形態1に記載したプロセス復元処理と同様である。

【0127】

次に、第2の実施の形態について実施例を以下に示す。

【0128】

図23はプロセス生成時のプロセス退避情報作成手段135の処理内容を示している。

【0129】

ユーザによりプログラムの実行要求がOSに指示されると、OSはプロセスを生成する。ここでは、ある一つのプロセスの名称としてプロセスAと呼ぶ。プロセスAが生成され、プロセスディスクリプタがプロセス実行キューに登録されることで初めてプロセスAが実行される。プロセス退避情報生成手段135は、プロセス実行キューに当該プロセスが登録される前に実行する。プロセスAが生成された時点で、プロセスディスクリプタが格納されているアドレスは、すでにOSが保持している(OS内の一時変数に生成したプロセスAのプロセスディスクリプタが格納されている)。よって、プロセスディスクリプタのアドレスを取得する処理は行わない。まず最初にプロセスAの退避情報を格納する領域をメモリ上に確保する(ステップS1501)。次に、プロセスAのプロセスディスクリプタを確保したプロセス情報退避領域125にコピーする(ステップS1502)。さらにプロセスディスクリプタに付随する他の構造体(ファイル構造体、ソケット構造体など)を同退避領域125にコピーする(ステップS1503)。次に、プロセスAのページテーブルを同退避領域125にコピーする(ステップS1504)。ページテーブルが置かれているアドレスは、プロセスディスクリプタ中のメモリディスクリプタに格納されているので、それを参照する。最後に同退避領域125を退避領域リストに登録し、退避領域サイズに使用した主記憶サイズを足す(ステップS1505、S1506)。

【0130】

図24にプロセス退避情報更新手段136の処理内容を示す。更新処理は、プロセス切り替えに関する処理がすべて終了し、次に実行されるプロセスのレジスタ情報がタスク状態セグメントから復元される前に行う。プロセスAのプロセス情報退避領域125の更新処理を行う前に、まずユーザから当該プロセスを再起動するかどうかの命令を受け取っているかを確認する。この命令はシステムコールとして実装されてもよいし、他の方法を使っても良い。ユーザから当該プロセスを再起動するとの指示があると(ステップS1601のYES)、更新処理としては、プロセスAのプロセス情報退避領域125にある再起動フラグ203を立てて、処理を終了する(ステップS1607)。ユーザから再起動が指示されていなければ(ステップS1601のNO)、以下のように処理を行う。

【0131】

まず、プロセスAのスタックポインタよりプロセスディスクリプタのアドレスを取得する(ステップS1602)。次に、退避領域124からプロセスAのプロセス情報退避領域125のアドレスを取得する。図2における退避領域124のデータ構造を参照するとわかるように、退避領域124は、各プロセスのプロセス情報退避領域125をリストで管理している。よって、一番先頭(リストの最初のプロセス情報退避領域125)のアドレスを取得し、つぎにリストを探索していくことで目的のプロセスのプロセス情報退避領域125のアドレスが取得できる。現在、アクセスしている退避領域124がどのプロセスのプロセス情報退避領域125かを確認するには、プロセス生成時点でコピーされているプロセスディスクリプタ内のプロセスIDを見ればよい。プロセスAの退避領域124のアドレスを取得できれば、次にプロセスAのプロセスディスクリプタと付随する構造体をプロセスAのプロセス情報退避領域125にコピーする(ステップS1604、S1605)。最後に、ページテーブルをコピーする(ステップS1606)。

【0132】

プロセス終了時のプロセス情報退避領域解放手段137の処理内容は、第2の実施の形態と同様である。まず、退避領域124のリストからプロセスAのプロセス情報退避領域125のアドレスを取得する。アドレスが取得できたら、プロセス情報退避領域125をリストから削除し、領域を解放する。その後プロセスの終了処理を行う。

【0133】

以上でOSの持つプロセス情報の退避処理がすべて完了する。このように計算機システムの運用中に退避情報を作成しておくことで、再起動時には終了処理を行わなくてすむ。よって、終了処理に伴う補助記憶装置へのアクセスの削減に伴って再起動時間の短縮が実現できる。

【産業上の利用可能性】

【0134】

本発明によれば、再起動時にユーザへのサービスを中断することがないので、長時間連続運用が求められる大規模アプリケーションサーバーといった用途に適用できる。また、通常のパーソナル情報処理装置上のOS、たとえばWindowsなどにおけるレジストリ書き換えによる再起動の際にも、ユーザの作業状況はそのままに、OSのみ再起動することにより書き換えたレジストリ情報を反映させるような用途として適用可能である。

【図面の簡単な説明】

【0135】

【図1】本発明の第1の実施の形態にかかる情報処理装置の構成を示すブロック図である。

【図2】本発明の第1の実施の形態における退避領域のデータ構造を示す図である。

【図3】本発明の第1の実施の形態の動作を示す流れ図である。

【図4】本発明の第1の実施の形態におけるブロック識別子を示す図である。

【図5】本発明の第1の実施の形態における主記憶初期化手段の処理を示す流れ図である。

【図6】本発明の第1の実施の形態の変形例の構成を示すブロック図である。

【図7】本発明の第1の実施の形態の変形例の動作を示す流れ図である。

【図8】本発明の第2の実施の形態にかかる情報処理装置の構成を示すブロック図である。

【図9】本発明の第2の実施の形態におけるプロセス退避情報作成手段の処理を示す流れ図である。

【図10】本発明の第2の実施の形態におけるプロセス退避情報更新手段の処理を示す流れ図である。

【図11】本発明の第2の実施の形態におけるプロセス情報退避領域解放手段の処理を示す流れ図である。

【図12】本発明の第3の実施の形態で利用する主記憶イメージを用いた高速起動方式を適用した情報処理装置の構成図である。

【図13】本発明の第3の実施の形態で利用する主記憶イメージを用いた高速起動方式を適用した情報処理装置におけるマッピングテーブル部のエントリ構成を示す図である。

【図14】本発明の第3の実施の形態で利用する主記憶イメージを用いた高速起動方式を適用した情報処理装置における保護情報部のエントリ構成を示す図である。

【図15】本発明の第3の実施の形態で利用する主記憶イメージを用いた高速起動方式を適用した情報処理装置の処理を示す流れ図である。

【図16】本発明の第3の実施の形態で利用する主記憶イメージを用いた高速起動方式を適用した情報処理装置の処理を示す流れ図である。

【図17】本発明の第3の実施の形態にかかる情報処理装置の構成を示すブロック図である。

【図18】本発明の第3の実施の形態にかかる情報処理装置における主記憶初期化手段の処理を示す流れ図である。

【図 1 9】本発明の第4の実施の形態にかかる情報処理装置の構成を示すブロック図である。

【図 2 0】本発明の第4の実施の形態にかかる情報処理装置における再起動フラグ設定手段の処理を示す流れ図である。

【図 2 1】本発明の第1の実施例におけるプロセス情報退避手段の処理を示す流れ図である。

【図 2 2】本発明の第1の実施例における主記憶初期化手段の処理を示す流れ図である。

【図 2 3】本発明の第2の実施例におけるプロセス退避情報作成手段の処理を示す流れ図である。

【図 2 4】本発明の第2の実施例におけるプロセス退避情報更新手段の処理を示す流れ図である。

【図 2 5】従来の情報処理装置の構成を示すブロック図である。

【符号の説明】

【0 1 3 6】

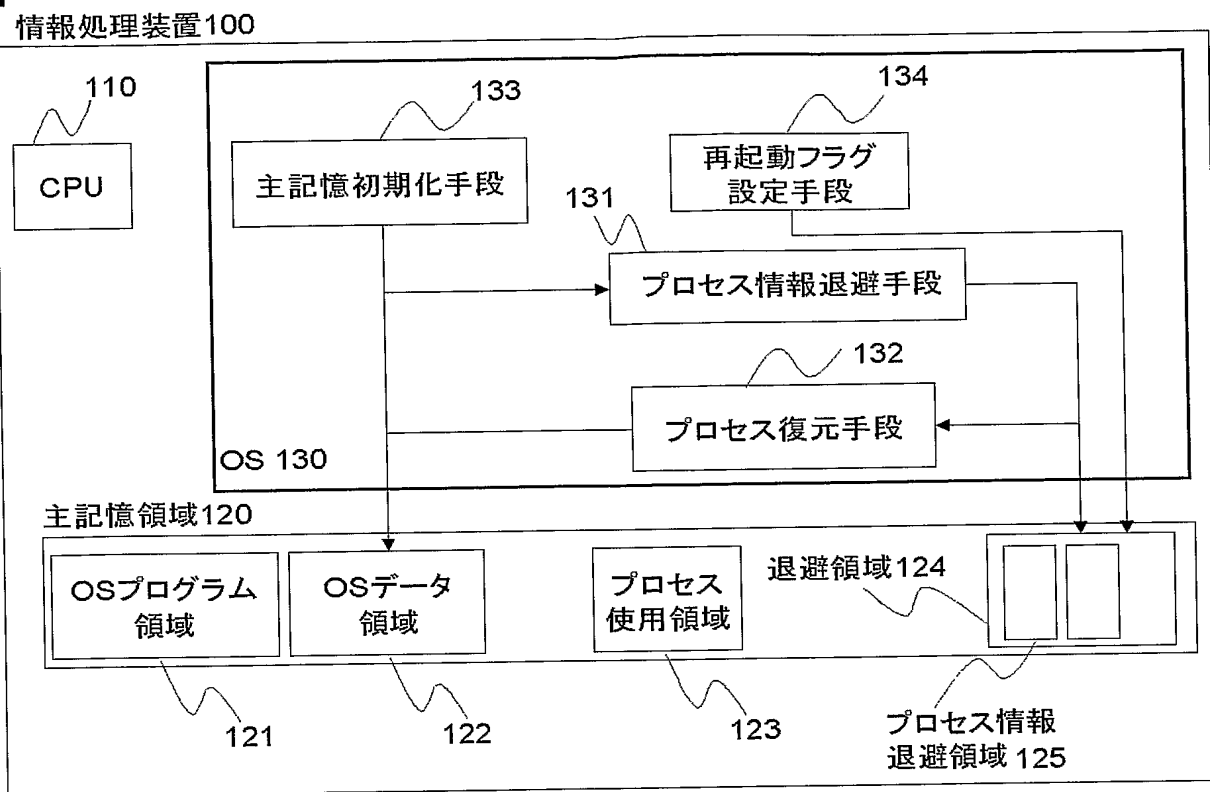
- 100、100a～100d 情報処理装置
- 101 主記憶装置
- 110 CPU
- 120 主記憶領域
- 121 OSプログラム領域
- 122 OSデータ領域
- 123 プロセス使用領域
- 124 退避領域
- 125 プロセス情報退避領域
- 126 主記憶部
- 130 OS
- 131 プロセス情報退避手段
- 132 プロセス復元手段
- 133 主記憶初期化手段
- 134 再起動フラグ設定手段
- 134a 再起動プロセスIDテーブル
- 134b 再起動設定手段
- 135 プロセス退避情報作成手段
- 136 プロセス退避情報更新手段
- 137 プロセス情報退避領域解放手段
- 140 不揮発性記憶部
- 150 データ処理装置
- 160 マッピングテーブル部
- 161 論理ブロックアドレス
- 162 物理ブロックアドレス
- 163 保護モード
- 170 保護情報部
- 171 論理ブロックアドレス
- 172 保護属性
- 180 マッピングテーブル初期化部
- 190 マッピングテーブル変更部
- 200 アドレス変換部
- 201 退避領域使用サイズ
- 202 プロセス表
- 203 ブロックテーブル

- 204 再起動フラグ
- 205 単方向リスト
- 210 ハードウェア初期化部
- 220 再起動設定受付手段
- 221 再起動プロセスIDテーブル
- 222 再起動フラグ設定手段
- 400 ブロック識別子
- 401 ブロック番号
- 402 使用フラグ

【書類名】 図面

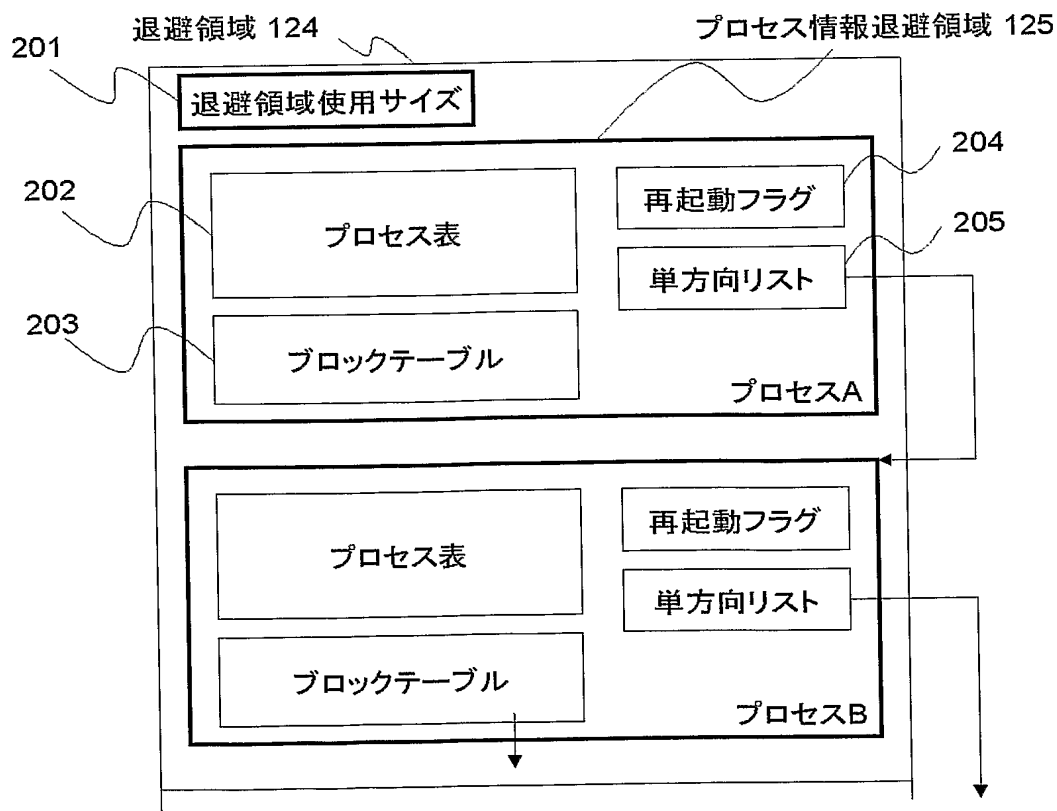
【図 1】

【圖 1】



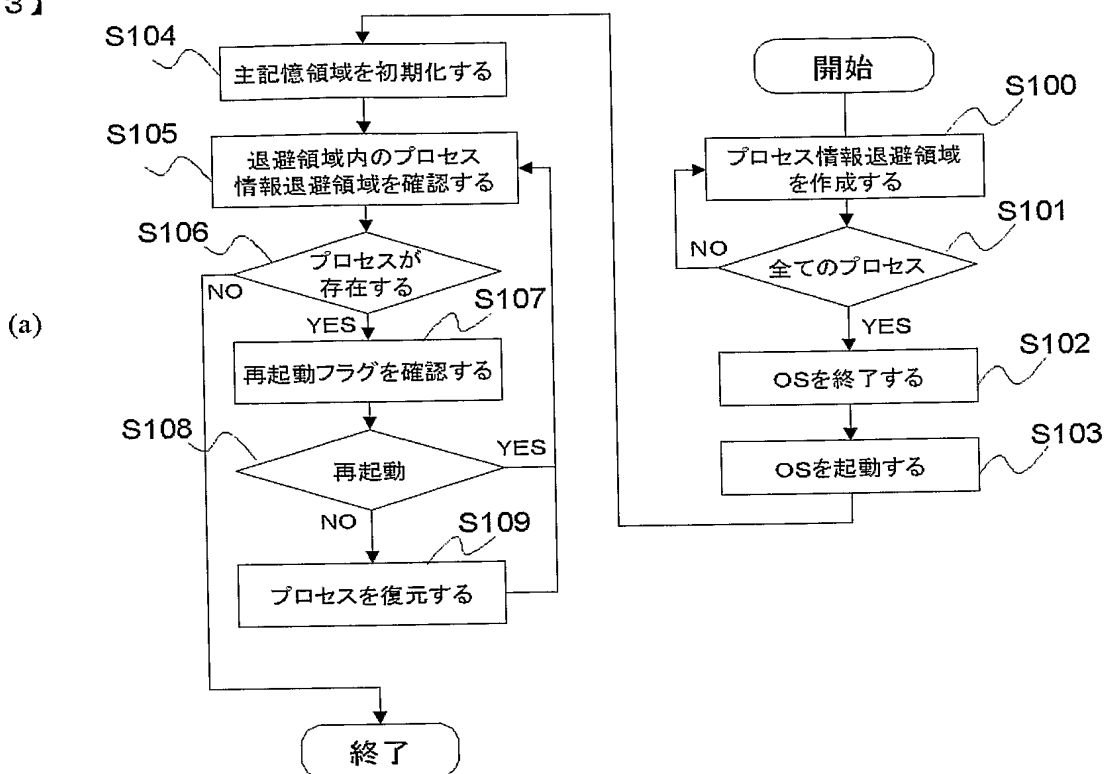
【図 2】

【圖 2】

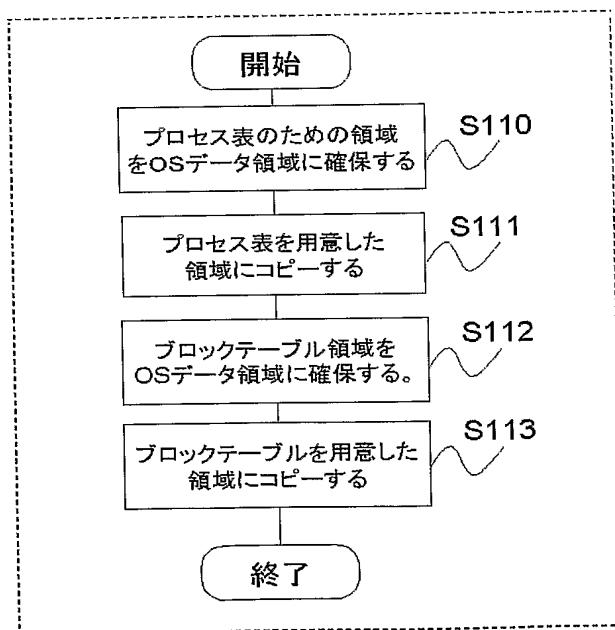


【図 3】

【図 3】

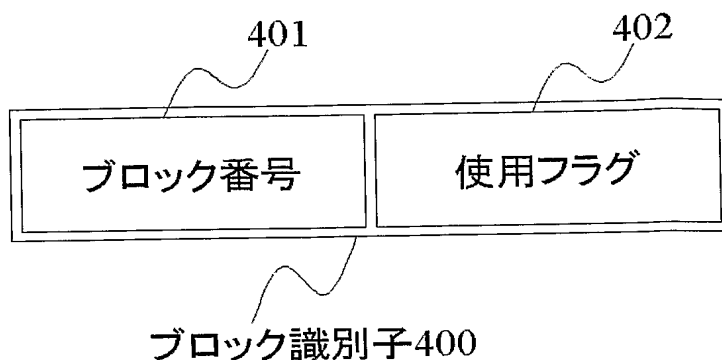


(b)

「プロセス復元」部分
詳細フローチャート

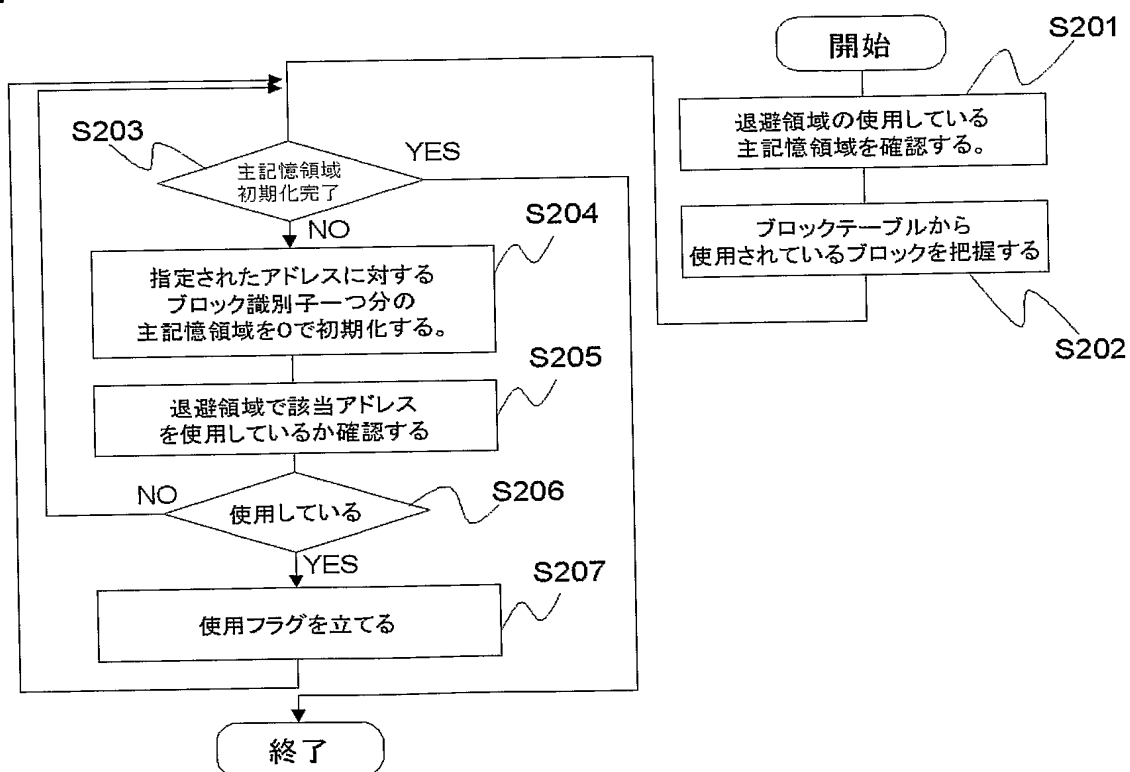
【図 4】

【図 4】



【図 5】

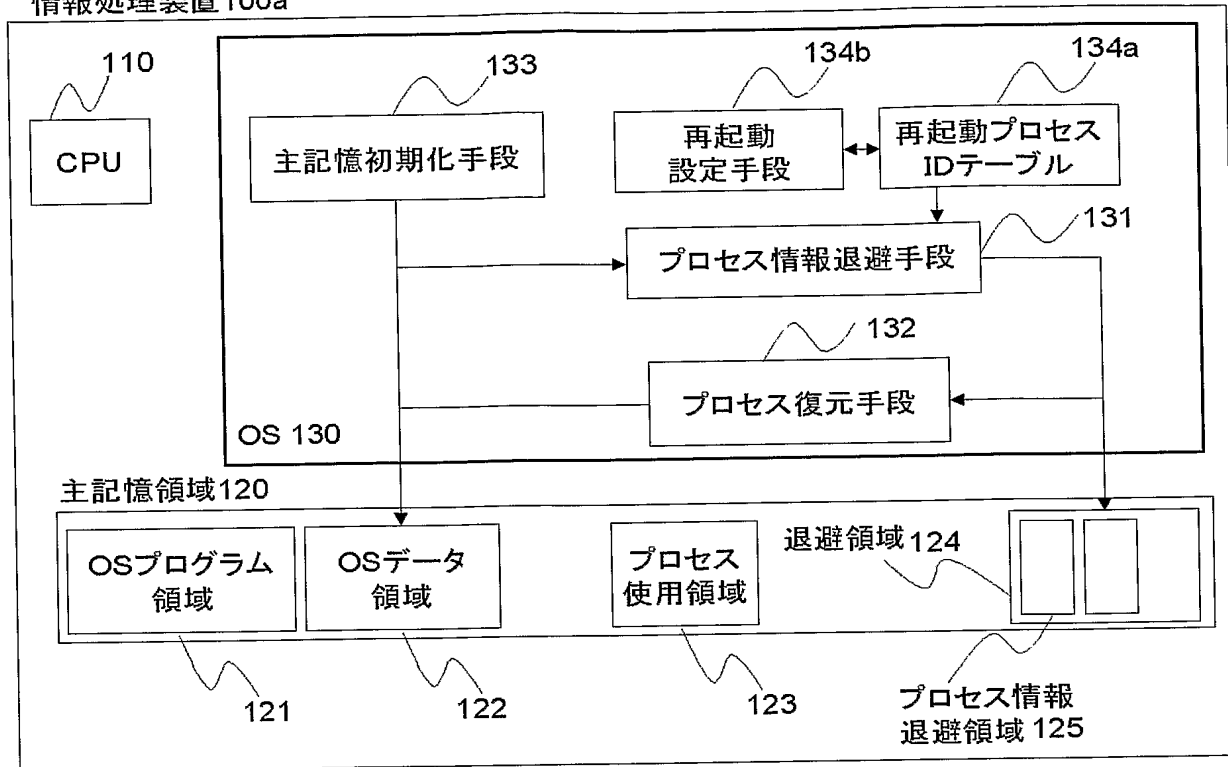
【図 5】



【図6】

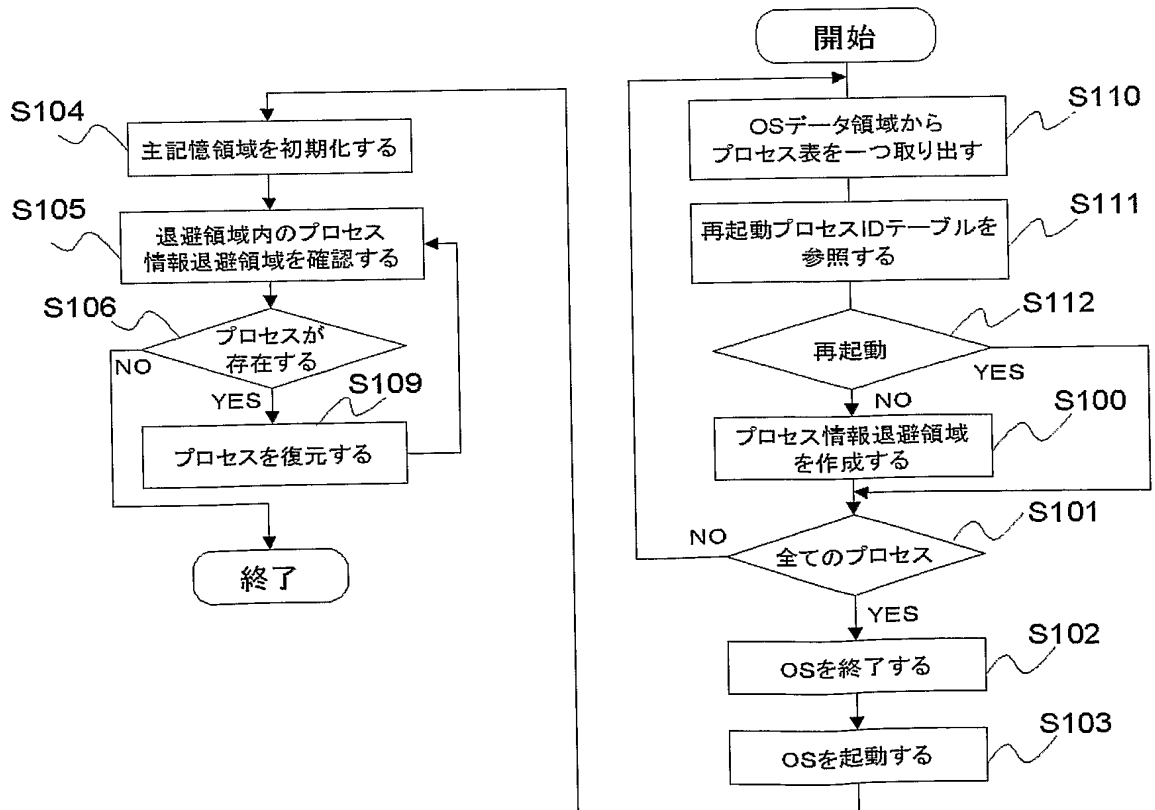
【図6】

情報処理装置100a



【図7】

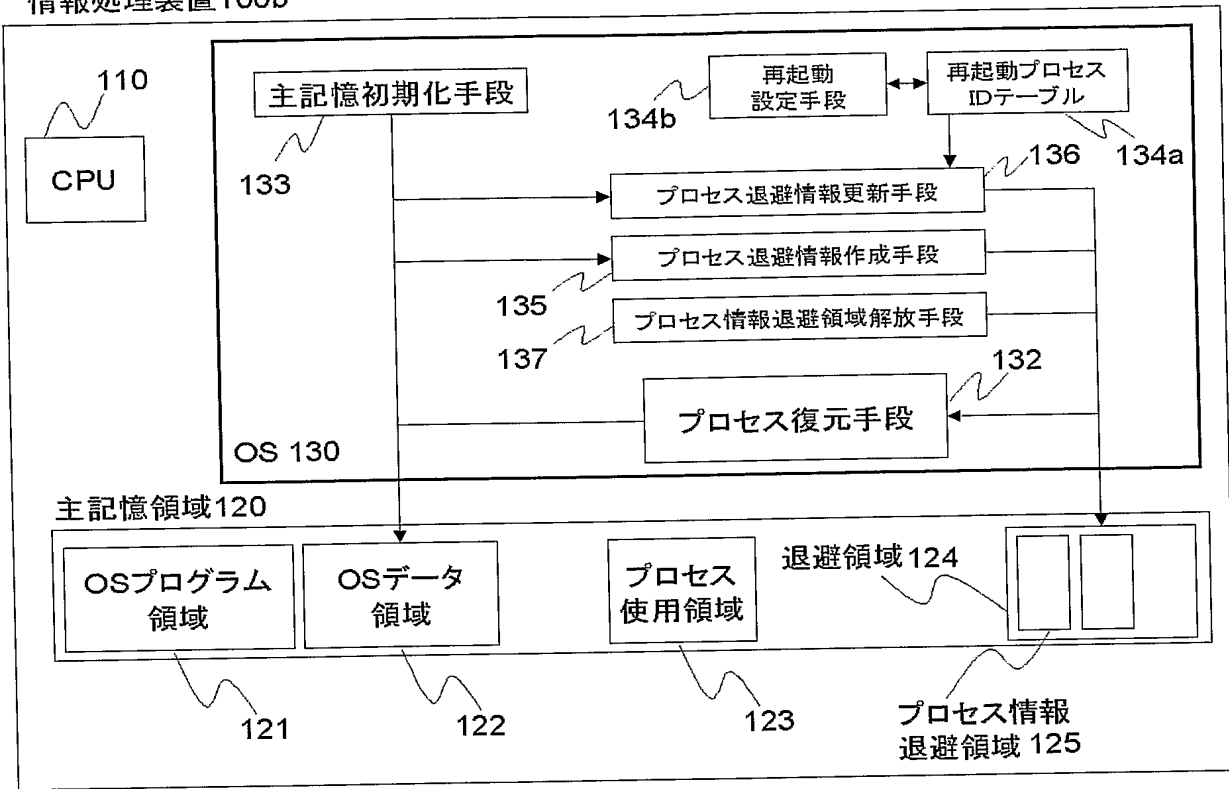
【図7】



【図 8】

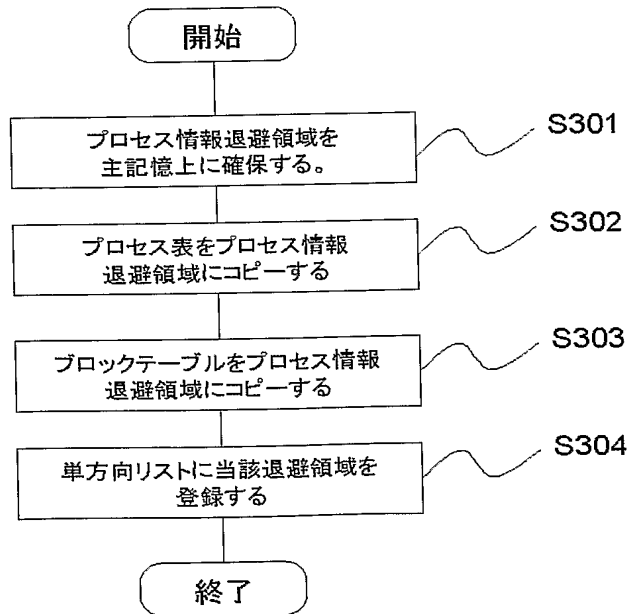
【図 8】

情報処理装置100b



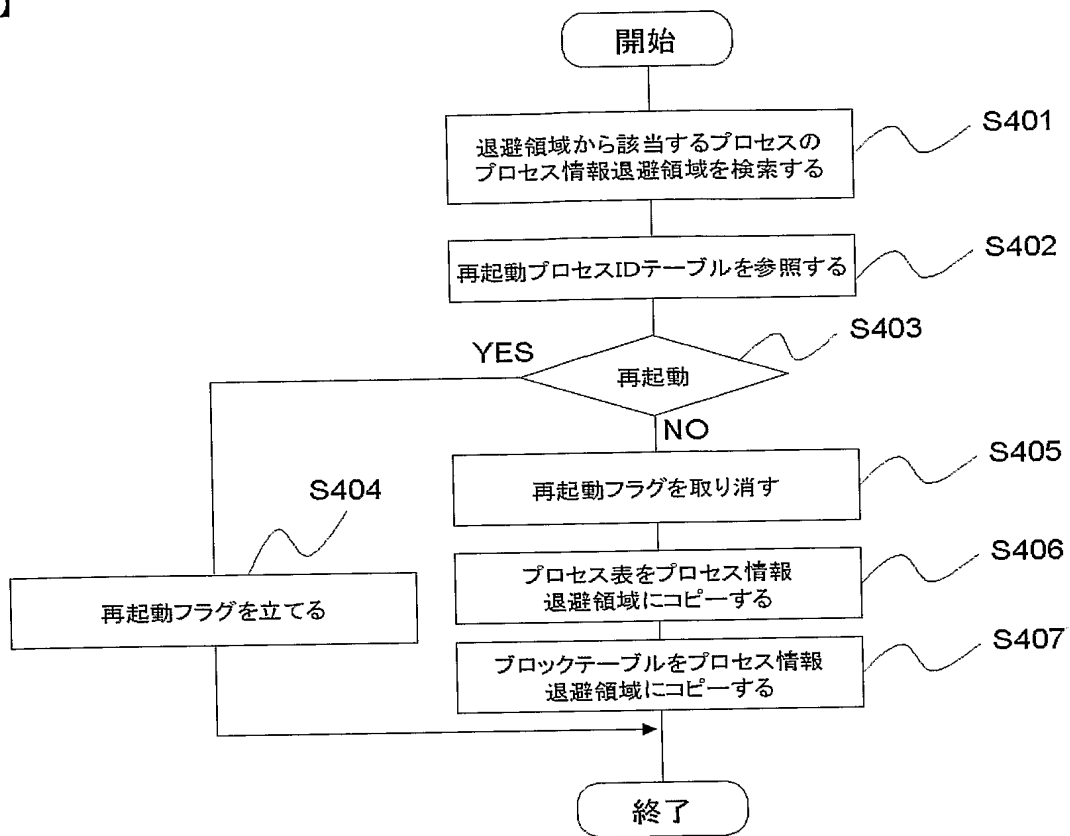
【図 9】

【図 9】



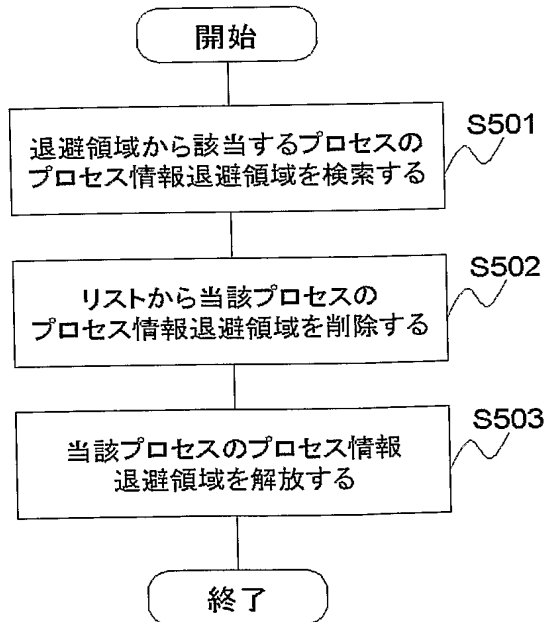
【図 10】

【図 10】



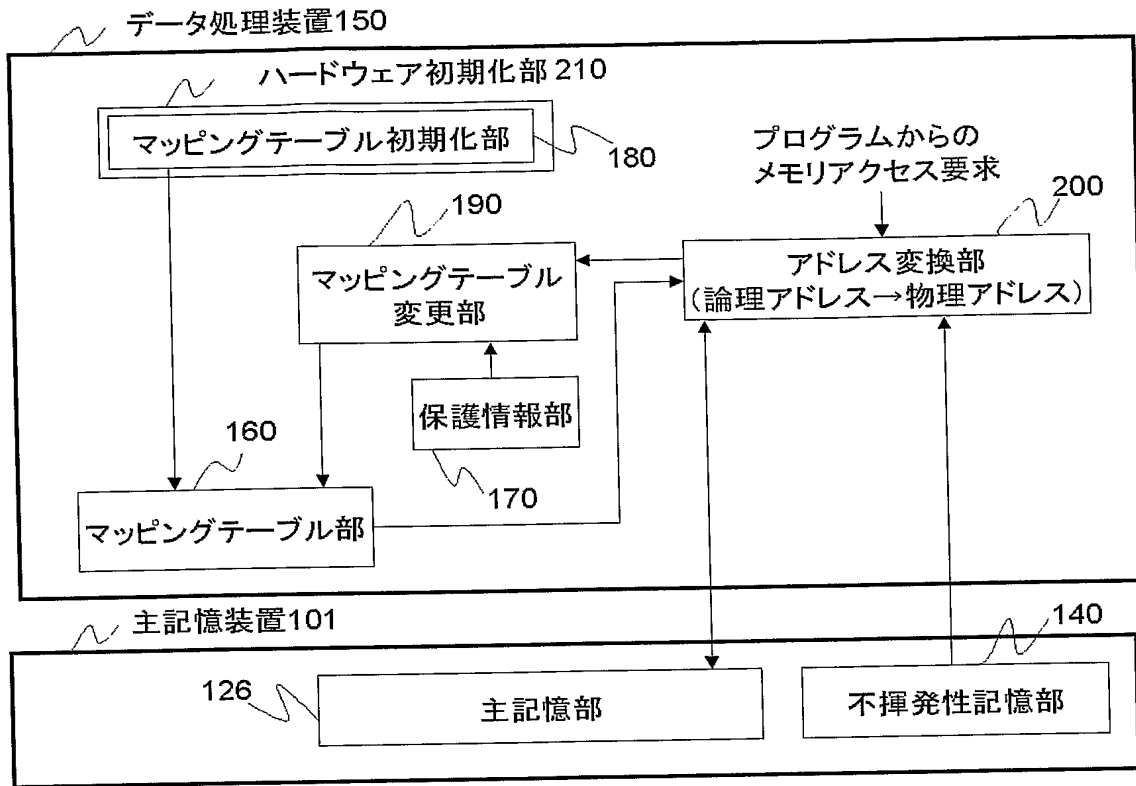
【図 11】

【図 11】



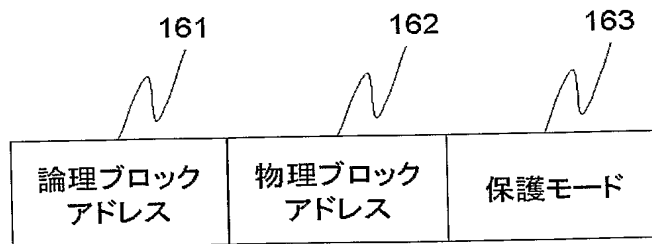
【図 12】

【図 12】



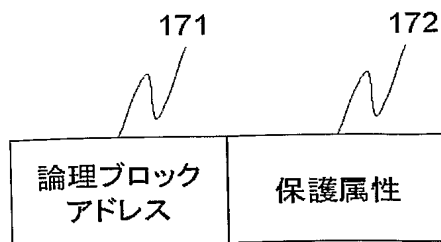
【図 13】

【図 13】



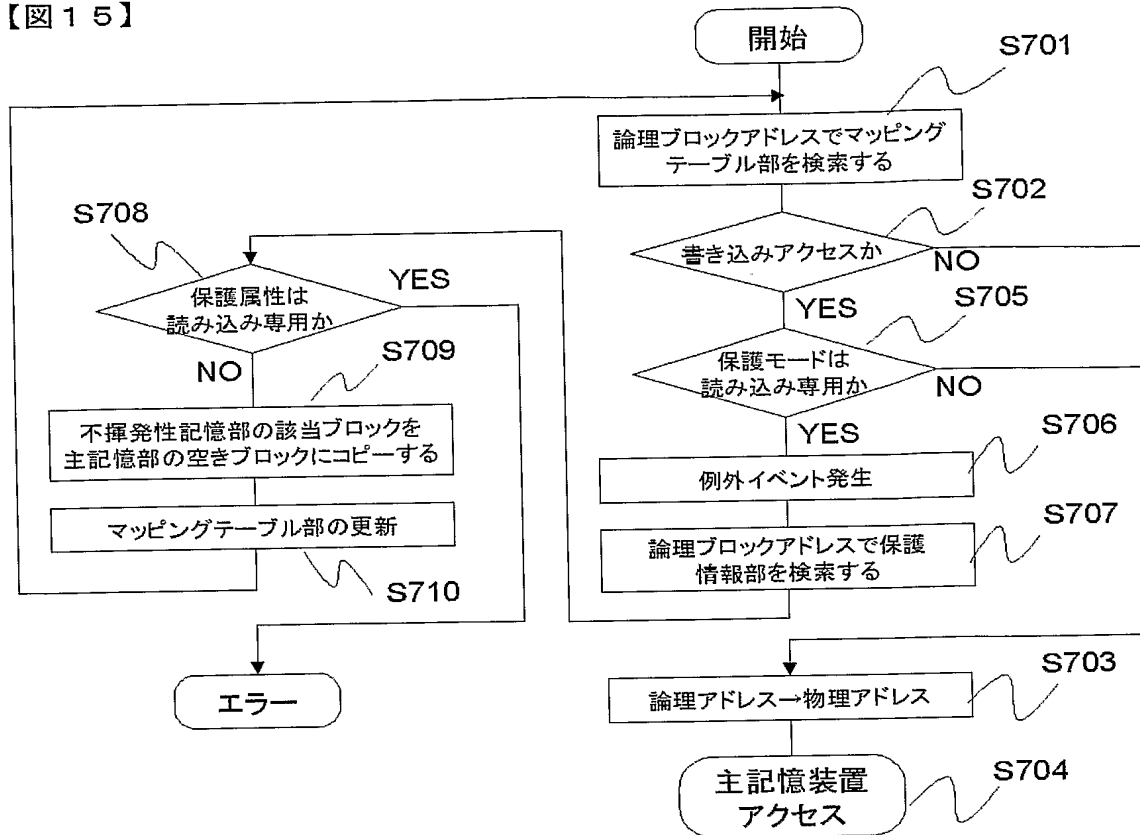
【図 14】

【図 14】



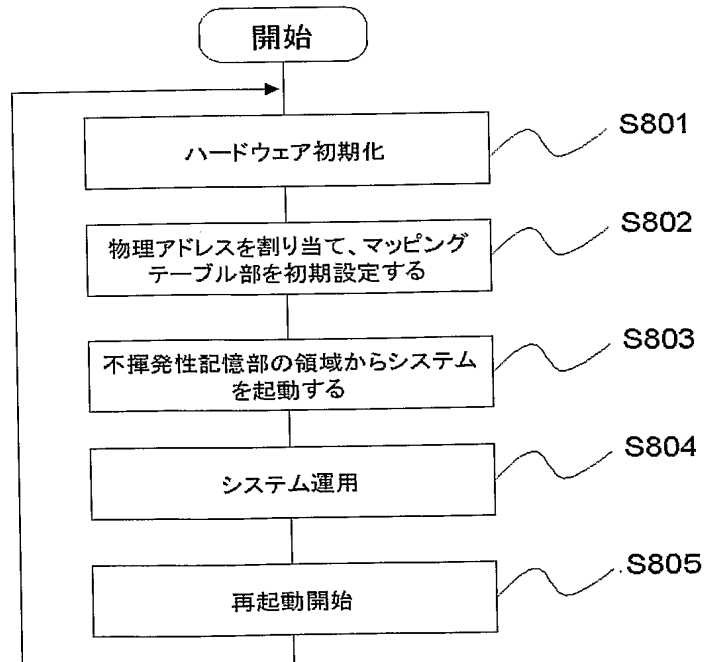
【図 15】

【図 15】



【図 16】

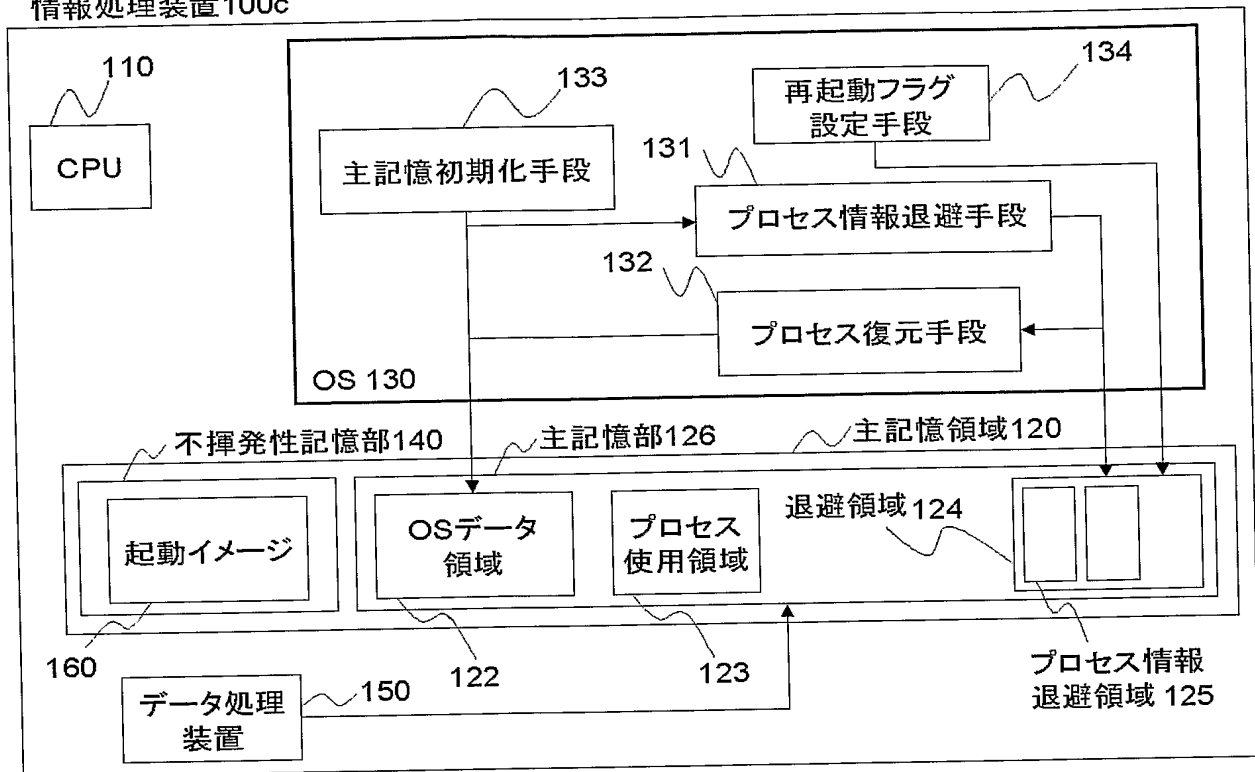
【図 16】



【図 17】

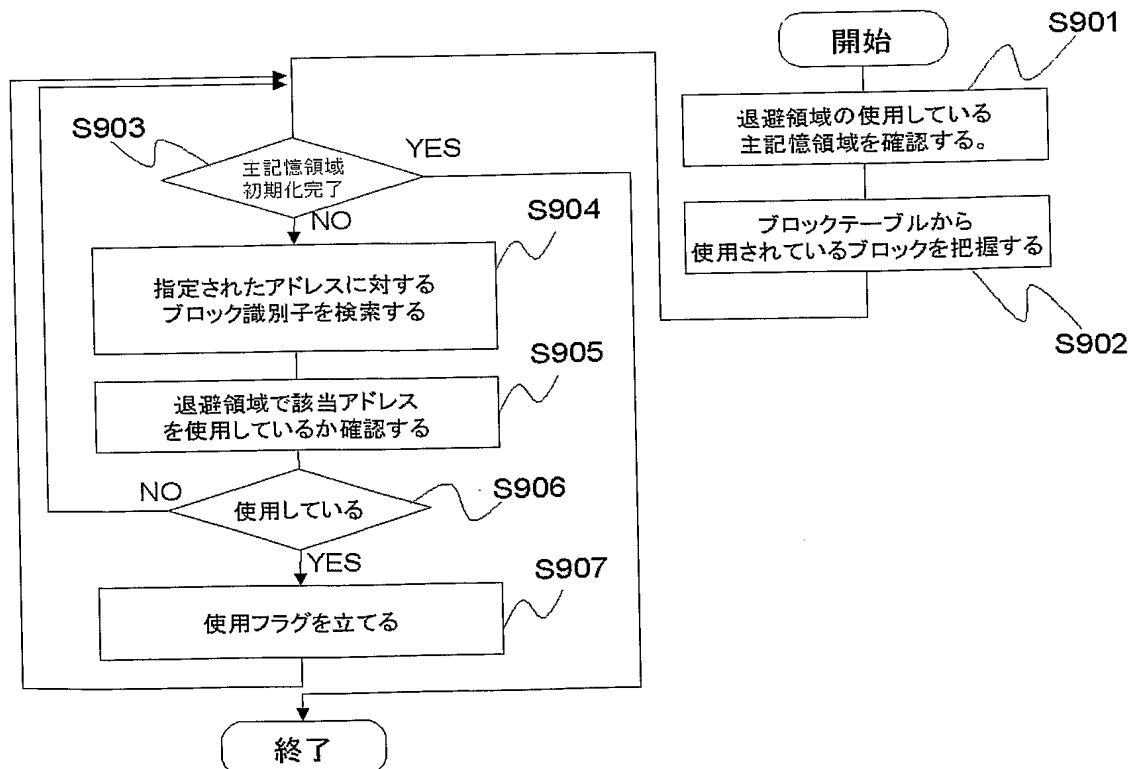
【図 17】

情報処理装置100c



【図 18】

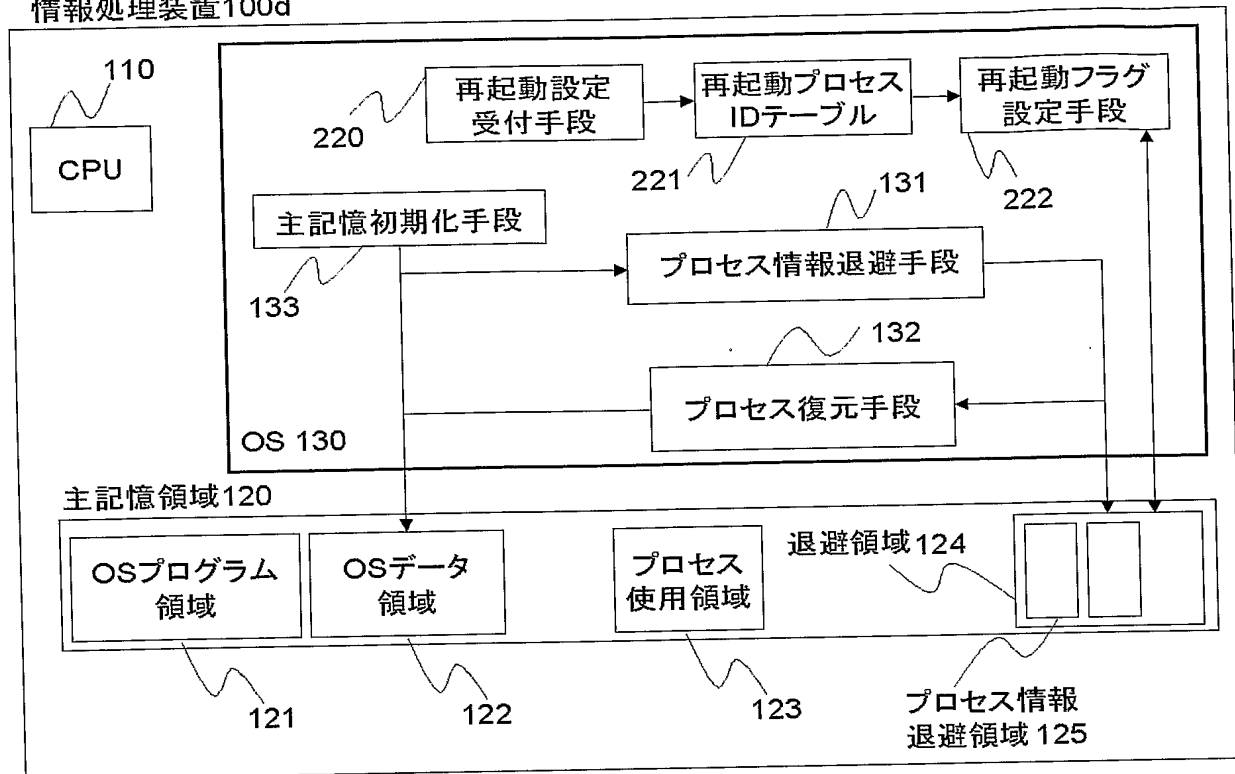
【図 18】



【図 19】

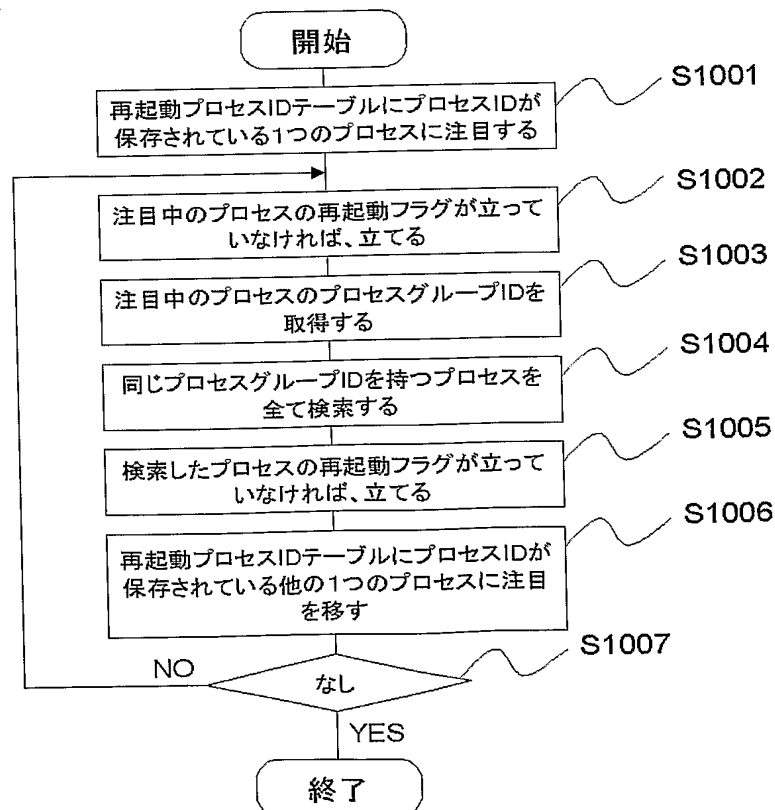
【図 19】

情報処理装置100d



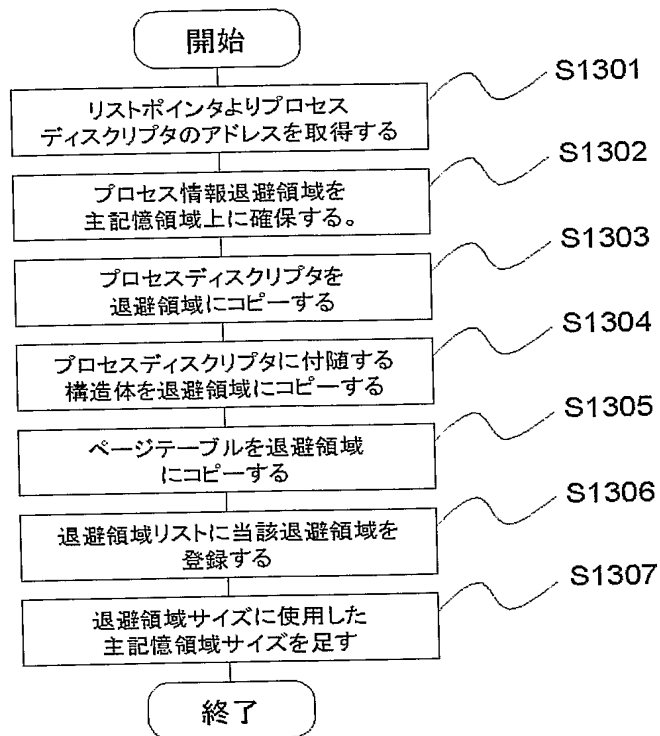
【図 20】

【図 20】



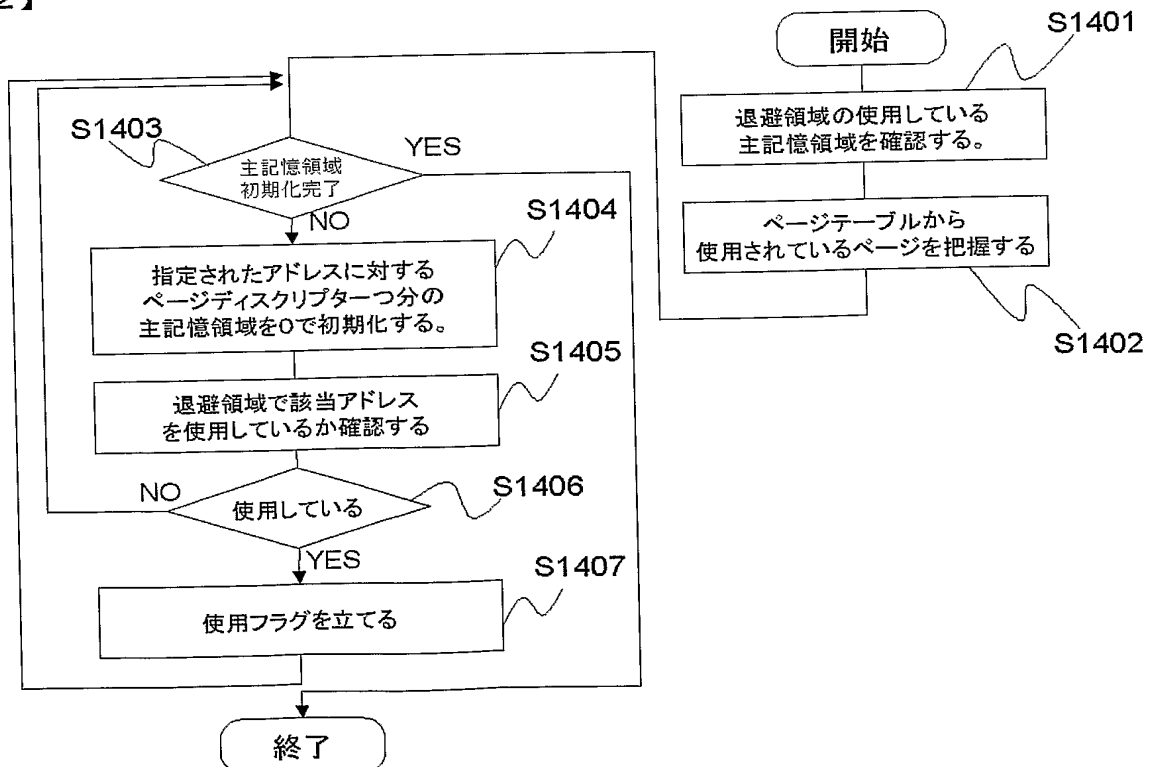
【図 2 1】

【図 2 1】

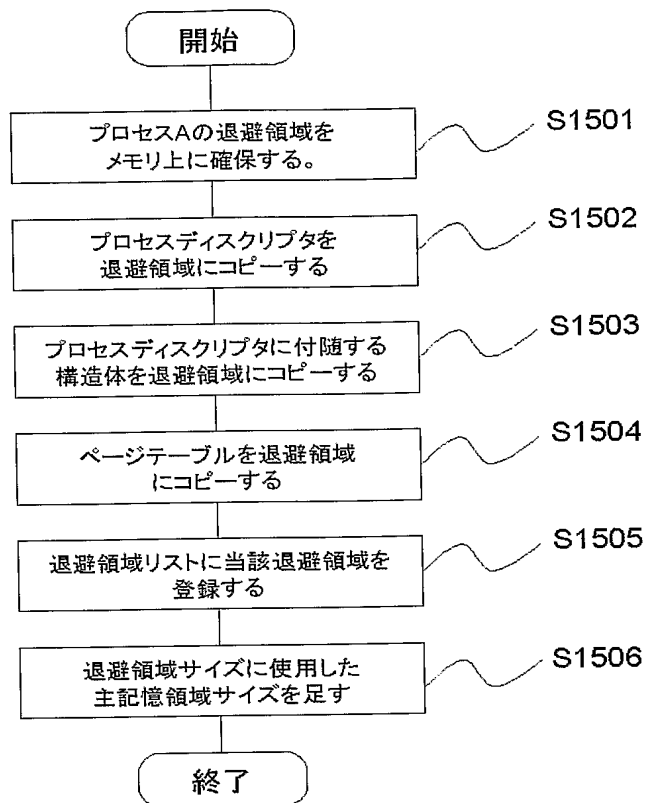


【図 2 2】

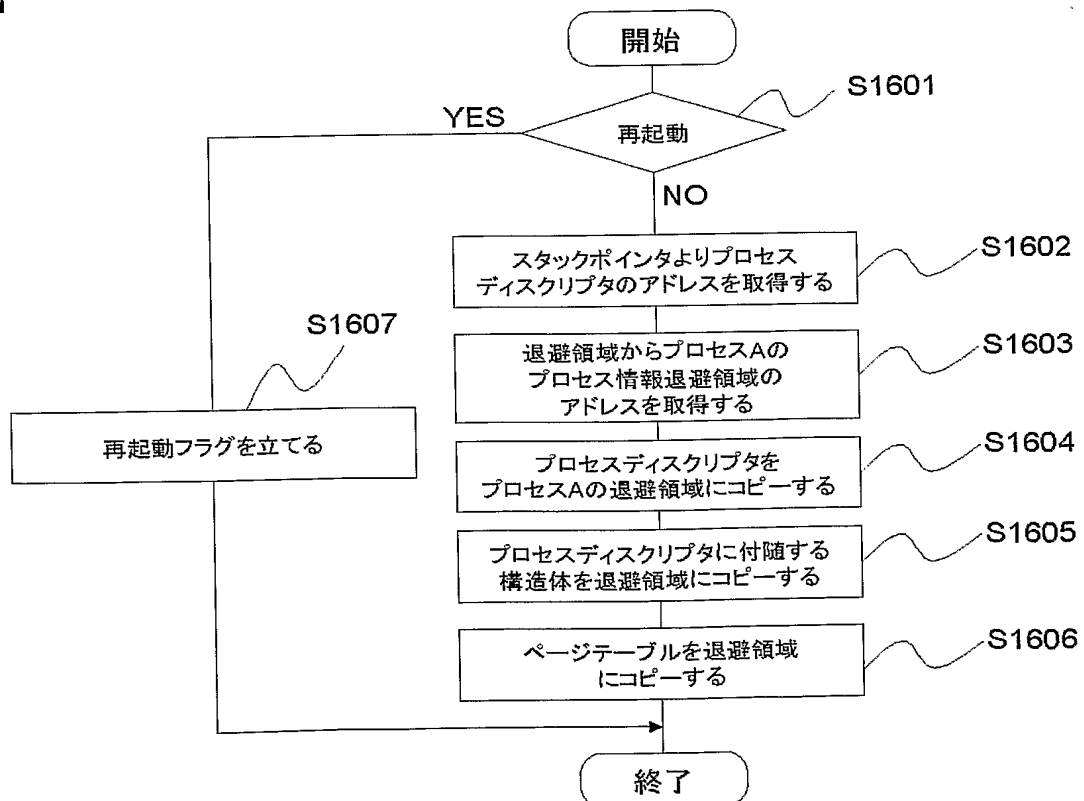
【図 2 2】



【図 2 3】
【図 2 3】

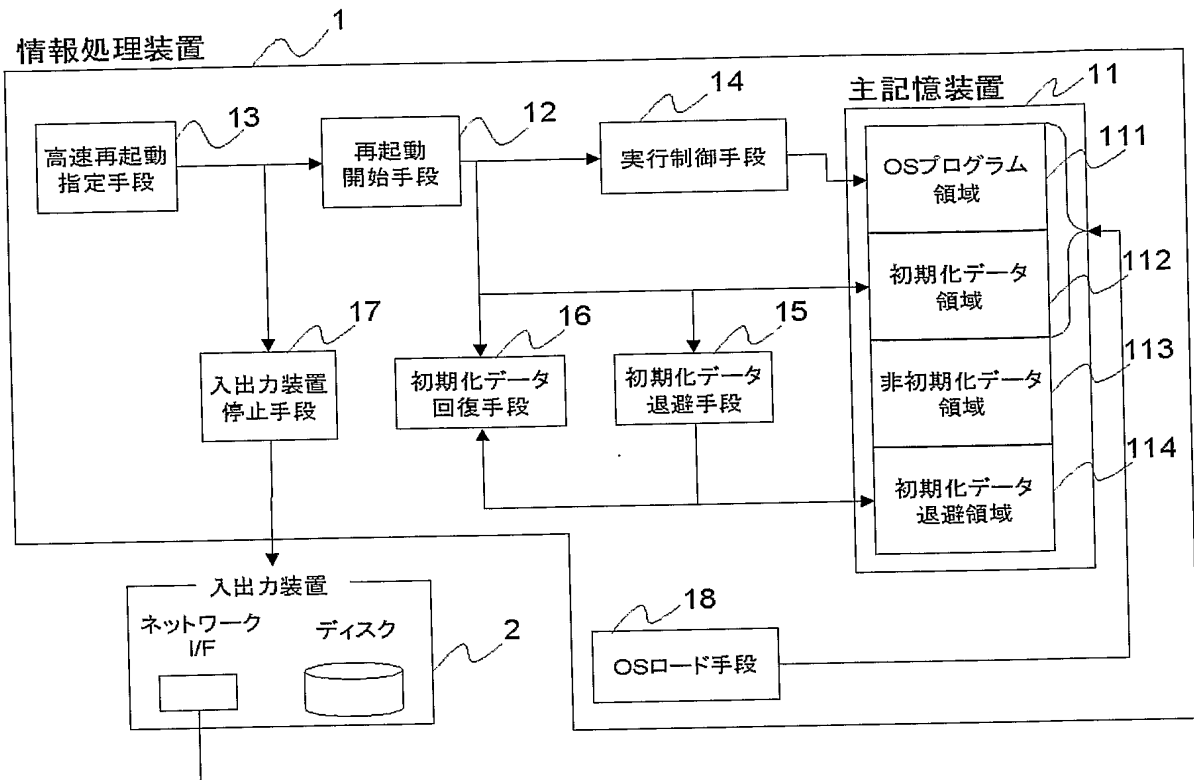


【図 2 4】
【図 2 4】



【図 25】

【図 25】



【書類名】 要約書

【要約】

【課題】 情報処理装置の高速再起動ならびに、再起動後におけるアプリケーションの継続運用を実現する。

【解決手段】 プロセス情報退避手段131と、プロセス復元手段132とを備え、アプリケーションの実行時にOS 130内に存在するプロセス表201やその他、プロセスを復元するために必要な情報を主記憶領域120上の退避領域124に複写しておき、再起動時には、アプリケーションの使用していた主記憶領域はそのままに、OSのみを初期化する。初期化の後に、退避しておいたプロセス情報(プロセス表やその他OSが管理していた情報)をOSデータ領域122に上書きするよう動作する。このような構成を採用し、再起動前のプロセスを再起動後も継続実行できる環境を提供する。

【選択図】 図1

特願 2 0 0 4 - 0 1 8 5 2 4

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 4 2 3 7]

1. 変更年月日

1 9 9 0 年 8 月 2 9 日

[変更理由]

新規登録

住 所

東京都港区芝五丁目 7 番 1 号

氏 名

日本電気株式会社